Current progress on the numerical simulation of detached flows around airplanes

C. Begue, Q.V. Dinh, B. Mantel, Jacques Périaux, G. Terrasson, B. Cardot, Fadi El Dabaghi, Frédéric Hecht, R. Munoz, C. Pares, et al.

INRIA

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél. (1) 39 63 55 11

Rapports de Recherche

N° 778

CURRENT PROGRESS ON THE NUMERICAL SIMULATION OF DETACHED FLOWS AROUND AIRPLANES

Catherine BEGUE, Quang Vinh DINH
Bertrand MANTEL, Jacques PERIAUX
Guillaume TERRASSON, Bernard CARDOT
Fadi El DABAGHI, Frédéric HECHT
Rafael MUNOZ, Carlos PARES
Olivier PIRONNEAU
El Montasser ABDALASS
Roland GLOWINSKI

DECEMBRE 1987

CURRENT PROGRESS ON THE NUMERICAL SIMULATION OF DETACHED FLOWS AROUND AIRPLANES.

Catherine Bègue , Quang Vinh Dinh, Bertrand Mantel,
Jacques Périaux, Guillaume Terrasson
(AMD-BA 78 Quai Carnot ST. Cloud France)
Bernard Cardot , Fadi El Dabaghi , Frédéric Hecht , Rafael Muñoz ,
Carlos Parés , Olivier Pironneau
(INRIA, 78153 LE CHESNAY, France)
El Montasser Abdalass (INRIA and ECL, 69130 Ecully, France)
Roland Glowinski (University of Houston, USA)

Abstract :

This work is carried out under a research project at the French Institut INRIA with the aim of implementing a computer program for airflows around vehicles of high velocity. We assume that the flow is described by a three-dimensional Navier-Stokes equations and we present here a boundary layer simulation model with a boundary condition which simulate the behaviour of logarithmic boundary sublayers. The theoretical results developed and some numerical results obtained have been presented.

DEVELOPPEMENTS RECENTS D'ALGORITHMES POUR LA SIMULATION NUMERIQUE DES ECOULEMENTS EN AERONAUTIQUE

Résumé :

Cet article résume certains progrès faits ces dernières années dans le domaine de la simulation numérique des équations de Navier-Stokes. Certaines méthodes présentées ici sont encore en cours de réalisation. Les recherches ont été menées conjointement à l'INRIA et aux AMD/BA tant en ce qui concerne le cas compressible que le cas incompressible. Toutefois, seul ce dernier cas est exposé ici. Les efforts ont porté sur la réalisation d'algorithmes performants, la prise en compte de modèles de turbulence, de conditions aux limites générales et des couches limites.

1. INTRODUCTION

Numerical simulations of flows around airplanes are still a challenge to the numerical analyst when the boundary layers separate because the full Navier-Stokes equations must be used and turbulences develop in the wakes of the airplanes. Furthermore airplanes have complex 3D shapes around which it is very difficult to build structured meshes. The purpose of this paper is to report on some methods to deal with these two difficulties at AMD-BA and INRIA when the Finite Element Method is used to approximate the Partial Differential Equations.

Faced with the complicated geometrical problem for complete aiplane discretization, it was decided in the 70's to choose the finite element method with tetraedra because it seemed very difficult to built a computer code and meshes that could deal with so different cases as aiplanes with/without external engines, air intakes, space shuttles.... Another advantage with the finite element method is that error estimates are available and so the limit of validity of the code is known. On the other hand the method is more expensive in computer memory and to some extend in computing time because sophisticated accelerators like multigrid iterations or reduced quadratures are harder to implement.

To deal with the second difficulty, turbulence, we have chosen to push the direct Navier-Stokes solver to its limits and to include turbulence modelling only in the near wake. This is because we are mainly interested in pressure drags and turbulence does not affect the airplane drag as much as it affects the heat transfer processes for the space shuttle . However the work is still in

progress because the classical turbulence models seem not to give satisfactory results for such flows. .

The paper is organized as follows. In the first part, we present the Finite Element scheme , the time discretization and the error estimates for the two basic schemes used in our labs. Then we discuss the various acceleration procedures under development and some tools to reduce the domain of integration.

2. GENERALITIES

2.1 A Finite Element Method for the Navier-Stokes Equations

Consider for instance the problem of simulating the flow around an airplane.

Since the dynamics of the problem is invariant by tranlation we may assume for simplicity that the vehicle is at rest while the surrounding reference frame moves at constant speed u_∞. Furthermore for computational reasons it is necessary to assume that the domain occupied by air around the aiplane is bounded. So we have 3 boundaries:

the boundary of the aiplane : Γ_1, .
the boundary which approximates infinity: Γ_2 ,

At moderate Reynolds number air may be assumed incompressible and newtonian; thus the equations for the velocity u (x,t) at point x and time t, and the pressure p(x,t) are the Navier-Stokes equations. It is possible to rescale the equations and write them as follows:

$$\mathbf{u}t + \mathbf{u}\nabla\mathbf{u} + \nabla p - \nu\Delta\mathbf{u} = 0 \text{ (conservation of momentum)} ; \tag{2.1}$$

$$\nabla.\mathbf{u} = 0 \text{ (conservation of mass)} . \tag{2.2}$$

One can see the rescaling of (2.1) and (2.2) in appendix B. Here ν is the reciprocal Reynolds number. The computational domain Ω is the region occupied by air in the domain bounded by the aiplane and the artificial boundary which approximate infinity; the diamater of the airplane is of order one and so is u_∞, the velocity at infinity.

Appropriate boundary conditions for this problem are

$$\mathbf{u}(x,0) = \mathbf{u}^o(x) \quad in \quad \Omega \text{ (initial conditions)} ; \tag{2.3}$$

$$\mathbf{u}(x,t) = \mathbf{g} \quad on \quad \Gamma \text{ (boundary condition)} . \tag{2.4}$$

g is given by the physics of the problem, i.e.

$$\mathbf{g} = 0 \quad on \quad \Gamma_1 \text{ (air does not slip on the aiplane)} ; \tag{2.5}$$
$$\mathbf{g} = \mathbf{u}_\infty \quad on \quad \Gamma_2 \text{ (free stream)} ;$$

However $\mathbf{u}^o$ may be chosen by the user. If the transient regime is not the main purpose of the computation then it is best to choose $\mathbf{u}^o$ so that the flow reaches its stationnary, or quasi-stationnary regime as fast as possible. Thus we have chosen to initialize the process with the inviscid solution of (2.1) and (2.2) which satisfies all the conditions except the no-slip boundary condition (2.5) ; we have then

$$\mathbf{u}^o(x) = \nabla\varphi(x); \tag{2.6}$$

where $\varphi(x)$ is the solution of the problem

$$\Delta\varphi = 0 \quad in \quad \Omega; \quad \frac{\partial\varphi}{\partial n} = \mathbf{g}.\mathbf{n}. \tag{2.7}$$

To approximate (2.1) , (2.2) and (2.7) by the Finite Element Method we must divide Ω into small elements and replace all the functions by their interpolate uh , p_h and φ_h. Interpolates are defined inside the elements from their values at the 'nodes' of the elements, i.e. if the values of the functions at the nodes are stored into the computer one is able to compute the values of theses functions at any other point by an interpolation formula. Thus to optimize the memory one ought to choose an element with few nodes; on the other hand the precision of the method is directly related to the number of nodes in each element and to the size of the element.

For instance, if Ω is divided into tetrahedra of average size h and if the nodes are the vertices of the tetrahedra then, to be consistent, one must use a linear interpolation approximation and the precision of the method will be of order h^2 at most. If the nodes are the vertices and the middle points on the edges then a quadratic interpolation approximation can be used and the precision will be of order h^3. A rapid calculation of the number of edges in a domain divided into tetrahedra will convince the reader that this number is usually much bigger than the number of vertices. Furthermore an investigation of the error between the exact value of φ at x and the interpolated value $\varphi_h(x)$ shows that

$$|\varphi(x) - \varphi_h(x)| \leq ||\varphi''||h^2 \text{ in the first case} ; \tag{2.8}$$

$$|\varphi(x) - \varphi_h(x)| \leq ||\varphi'''||h^3 \text{ in the second case .} \tag{2.9}$$

Thus if the function which is approximated has large derivatives the error will be large and the second formula may even turn to give bigger errors though it is theoreticaly more precise in terms of number of elements (i.e. in terms of h). For fluids this is the case. The vorticity is usually larger that the velocity and their derivatives even more, so experience has shown that it is not feasible to look for high order elements.

In the case of aiplane aerodynamics the computational domain is large and the problem is 3 dimensionnal. Therefore, computer memory is one of the major issue. This is again another argument in favor of an element with few nodes.

For equation (2.7) the solution is simple: one can choose the simplest element: the P^1 element i.e. φ is approximated by a piecewise affine function φ_h from its values at the vertices of the tetrahedra . So if we denote by Φ_h the space of such continuous piecewise affine functions, (2.7) is approximated by

$$\int_\Omega \nabla\varphi_h.\nabla w_h dx = \int_\Gamma \mathbf{g}.\mathbf{n} w_h d\gamma \quad \forall w_h \in \Phi_h; \tag{2.10}$$

By writing that φ_h is interpolated from its values at the vertices one finds that (2.10) is equivalent to a linear system

$$\mathbf{A}\Phi = \mathbf{G}. \tag{2.11}$$

Since A is a very large matrix one must be very careful to store only its non zero elements . This puts some restriction on the methods of solution of (2.11); in fact, on a non uniform triangulation one is essentially forced to use an iterative method. It is common knowledge that one of the best iterative method for the discrete Neumann problem (2.11) is the conjugate gradient method with some kind of preconditionning. Here an incomplete Cholesky factorisation is used to precondition the conjugate gradient algorithm if the core memory is suffisant , else **SSOR** is used (See for example Glowinski-Periaux-Pironneau[1]). Now we turn to the approximation of (2.1) and (2.2).

We use the variational form in the space of divergence free function [2], [3] ; if (.,.) is the $L^2(\Omega)^3$ scalar product we have :

$$(\mathbf{u}_{,t} + \mathbf{u}\nabla\mathbf{u}, \mathbf{v}) + \nu(\nabla\mathbf{u}, \nabla\mathbf{v}) = 0 \quad \forall \mathbf{v} \in \mathbf{J}_o(\Omega) \tag{2.12}$$

$$\mathbf{u} - \mathbf{u}_\Gamma \in J_o(\Omega); \mathbf{u}(x,0) = \mathbf{u}^o(x) \quad in \quad \Omega. \tag{2.13}$$

$$\mathbf{J}_o(\Omega) = \{\mathbf{v} \in H_o^1(\Omega)^3 : \nabla.\mathbf{v} = 0\} \tag{2.14}$$

where $\mathbf{u}_\Gamma$ is a divergence free extention of $\mathbf{g}$ inside Ω. Following [5] (see also [6]) we discretize (2.12) by replacing (2.14) by

$$\mathbf{J}_{oh} = \{\mathbf{v}_h \in \mathbf{v}_{oh} : (\nabla.\mathbf{v}_h, q_h) = 0 \quad \forall q_h \in Q_h\} \tag{2.15}$$

where V_{oh} and Q_h are suitable approximations of $(H_o^1(\Omega))^3$ and $L^2(\Omega)/R$ respectively. Among the simplest couples (V_{oh}, Q_h) are the so-called (cf. [19]) P^1 *bubble*$/P^1$ element :

V_h is the space of continuous piecewise polynomial vectors on a triangulation of Ω made of tetraedra ; on each tetraedron the polynomials must be of the form

$$\Sigma\alpha_i\lambda_i + \beta\lambda_1\lambda_2\lambda_3\lambda_4$$

where $\{\lambda_i\}$ are the barycentric coordinates with respect to the 4 vertices of the tetraedron. Thus, α_i is the value of the vector on the ith vertex and $\beta/81 + \Sigma\alpha_i/3$ is the value at the center of the tetraedron.

V_{oh} is the subset of V_h of vectors which are zero on the boundaries. Its dimensions is 3 times the number of vertices not on the boundaries plus 3 times the number of elements.

Q_h is the space of continuous piecewise affine functions on the triangulation, up to a constant ; its dimension is the number of vertices minus 1.

Then (2.12) is approximated by

$$(\mathbf{u}_{h,t} + \mathbf{u}_h\nabla\mathbf{u}_h, \mathbf{v}_h) + \nu(\nabla\mathbf{u}_h, \nabla\mathbf{v}_h) = 0, \quad \forall\mathbf{v}_h \in J_{oh,}, \tag{2.16}$$

$$\mathbf{u}_h - \mathbf{u}_{\Gamma h} \in J_{oh}.. \tag{2.17}$$

Error estimates can be found in [7] :

$$\|u - u_h\|_o \leq C\|u.\nabla u\|_o \frac{h^2}{\nu} \tag{2.18}$$

$$(\int_0^T \|u - u_h\|_1^2 dt)^{\frac{1}{2}} \leq C\|u.\nabla u\|_o \frac{h}{\nu} \tag{2.19}$$

where C is a numerical constant independant of u and ν ($\|.\|_o$ and $\|.\|_1$ are the L^2 and H^1 norm respectively).

The pressure p_h can be recovered as the Lagrange multiplier of the constraint in (2.15) and we have also

$$\|p - p_h\|_o \leq C\|u\nabla u\|_o \frac{h}{\nu} \tag{2.20}$$

2.2. Semi implicit Lagrangian/Eulerian characteristic time scheme.

To discretize (2.16) with respect to time, we use a semi implicit scheme based on the fact that if D/Dt denotes the particles derivative with the velocity field v , we have by definition

$$u_{h,t} + v\nabla u_h = \frac{Du_h}{Dt} \cong \frac{[u_h(x)^{n+1} - u_h^n(x - vk)]}{k} \tag{2.21}$$

here u_h^n is an approximation of u_h at time nk, where k denotes the time step Δt.

Let $X_h^n(x)$ be an approximation at τ = nk of the solution of

$$\frac{dX}{d\tau} = u_h^n(X_h, \tau); \quad X((n+1)k) = x \tag{2.22}$$

then (2.16) is replaced by

$$(u_h^{n+1} - u_h^n o X_h^n, v_h) + k\nu(\nabla u_h^{n+1}, \nabla v_h) = 0, \quad \forall v_h \in J_{oh}, \tag{2.23}$$

$$\mathbf{u}_h^{n+1} - \mathbf{u}_{\Gamma h} \in J_{oh}. \tag{2.24}$$

It was shown in [8], [9] that this scheme is unconditionnally stable and $O(h + \Delta t)$ for the L^2 norm. Furthermore this estimate holds even for $\nu = 0$. However (2.23) has a term which requires numerical quadrature :

$$(\mathbf{u}_h^n o X_h^n, \mathbf{v}_h) \cong \Sigma \mathbf{u}_h^n(X_h^n(\xi^i))\mathbf{v}_h(\xi^i)\pi_i \tag{2.25}$$

where ξ^i are the Gauss points and π_i the Gauss coefficients. With (2.25), no error estimates are available. Numerically, however experience shows that (2.21) is very reliable and it is one of the best method to solve the advection equation although, quadrature is a sensitive point of the method when u k/h is small.

2.3 Solution of the Stokes subproblems by the Uzawa/Conjugate gradient algorithm.

To solve (2.23) at each n we use Uzawa's algorithms on the pressure with preconditionned conjugate gradient (see [10]). A basis for V_{oh} and another one for Q_h are chosen ; V denotes the components of $\mathbf{u}_h^{n+1}$ on the basis of V_{oh} and P the Lagrange multiplier of the constraint in J_{oh} on the basis of Q_h , then (2.23), (2.24) is a linear systems of the form :

$$AV + BP = F, B^T V = 0; \tag{2.26}$$

where A_{ij} is $(\mathbf{v}^i, \mathbf{v}^j) + \nu k(\nabla \mathbf{v}^i, \nabla \mathbf{v}^j)$ if $\mathbf{v}^i$ is the basis of V_{oh}. Equivalently (2.26) is also

$$B^T A^{-1} BP = B^T A^{-1} F, \tag{2.27}$$

$$V = A^{-1}(F - BP). \tag{2.28}$$

Uzawa's algorithm amounts to solve (2.27) by a preconditionned conjugate gradient. As shown by Cahouet [11] a good preconditionning is the Neumann problem for the operator $(\nu I - k\Delta)$. Thus each time step of (2.23) will cost one computation of (2.25) (solving (2.22) at each Gauss point) and one computation of (2.27) i.e. 4n solution of a Laplace equation where n is the number of conjugate gradient iterations used in (2.27) (see Figure 2.3.1).

There are still plenty of room for improvements in the code ; for example a second order approximation could be used in (2.21) and/or a better preconditionning for the Neumann problems ; but the major improvement may come from the reduction of the size of the computation as we shall see below.

2.4 Solution of the Stokes problem by a boundary influence matrix method

We consider the Navier-Stokes problem described by (2.1)-(2.5). Using operator splitting techniques for time integration, we are able to decouple non linearity and incompressibility in the Navier-Stokes equations. Concerning incompressibility, we have to solve at each full step $(n\Delta t, (n+1)\Delta t)$ a general family of linear elliptic systems of the following type :

$$\begin{aligned} \alpha \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = f \quad in \quad \Omega, \\ \nabla . \mathbf{u} = 0 \quad in \quad \Omega, \\ \mathbf{u} = \mathbf{g} \quad on \quad \Gamma \, , (\text{with} \int_\Gamma \mathbf{g}.\mathbf{n} d\Gamma = 0), \end{aligned} \tag{2.29}$$

with $\alpha \cong 1/\Delta t$.

Following the Glowinski-Pironneau Stokes solver described in [12], we define the boundary operator A, as follows :

-i) To $\mu \in H^{-1/2}(\Gamma)$, we associate p_μ as the unique solution in

$$H(\Omega;\Delta) = \{q | q \in L^2(\Omega), \quad \Delta q \in L^2(\Omega)\} \tag{2.30}$$

of the Dirichlet problem

$$\Delta p_\mu = 0 \quad in \quad \Omega, \quad p_\mu = \mu \quad on \quad \Gamma. \tag{2.31}$$

-ii) To p_μ we associate $\mathbf{u}_\mu$ as the unique solution in $(H_0^1(\Omega))^3$ of the elliptic system

$$\alpha \mathbf{u}_\mu - \nu \Delta \mathbf{u}_\mu = -\nabla p_\mu \quad in \quad \Omega, \mathbf{u}_\mu = 0 \quad on \quad \Gamma. \tag{2.32}$$

-iii) To $\mathbf{u}_\mu$ we associate ψ_μ as the unique solution of $H_0^1(\Omega)$ of the Dirichlet problem

$$-\Delta\psi_\mu = \nabla.\mathbf{u}_\mu \quad in \quad \Omega, \quad \psi_{\dot{u}\mu} = 0 \quad on \quad \Gamma. \tag{2.33}$$

- iv) Finally, we define A by

$$A\mu = -\partial\psi_\mu/\partial\mathbf{n}|_\Gamma. \tag{2.34}$$

The operator A is an isomorphism from $H^{-1/2}(\Gamma)$ onto $H^{1/2}(\Gamma)$ where $H^{1/2}(\Gamma)$ and $H^{-1/2}(\Gamma)$ are respectively subsets of $H^{1/2}(\Gamma)$ and $H^{-1/2}(\Gamma)$ [12].

For the problem (2.29), we suppose $\mathbf{f} \in L^2(\Omega)^3$ and $\mathbf{g} \in (H^{1/2}(\Gamma))^3$, and we shall denote λ, the trace of p.

Let's define now $p_o \in H_0^1(\Omega)$, $\mathbf{u}_o \in (H^1(\Omega))^3$ and $\psi_o \in H^2(\Omega) \cap H_0^1(\Omega)$ by

$$\Delta p_o = \nabla.\mathbf{f} \quad in \quad \Omega, \quad p_o = 0 \quad on \quad \Gamma.$$

$$\alpha\mathbf{u}_o - \nu\Delta\mathbf{u}_o = \mathbf{f} - \nabla p_o \quad in \quad \Omega, |\mathbf{u}_o = \mathbf{g} \quad on \quad \Gamma. \tag{2.35}$$

$$-\Delta\psi_o = \nabla.\mathbf{u}_o \quad in \quad \Omega, \quad \psi_o = 0 \quad on \quad \Gamma.$$

we have $\partial\psi_o/\partial\mathbf{n}\,|_\Gamma \in H^{1/2}(\Gamma)$ and the pressure trace λ verifies the following ;

$$A\lambda = \frac{\partial\psi_o}{\partial\mathbf{n}}|_\Gamma. \tag{2.36}$$

Define now p , u andψ by :

$$p = p - p_o, \quad u = \mathbf{u} - \mathbf{u}_o, \quad \psi = -\psi_o.$$

we have then :

$$\Delta p = 0 \quad in \quad \Omega, \quad p = \lambda \quad on \quad \Gamma.$$

$$\alpha u - \nu\Delta u = -\nabla p \quad in \quad \Omega, \quad u = 0 \quad on \quad \Gamma. \tag{2.37}$$

$$-\Delta\psi = \nabla.u \quad in \quad \Omega, \quad \psi = 0 \quad on \quad \Gamma.$$

From the definition of operator A, it follows clearly that :

$$A\lambda = -\frac{\partial\psi}{\partial \mathbf{n}}|_\Gamma = \frac{\partial\psi_o}{\partial n}|_\Gamma,$$

From the above result, solving the Stokes problem (2.29), is equivalent to solving the following cascade of functional equations :

$$\begin{aligned} \Delta p_o = \nabla.\mathbf{f} \quad in \quad \Omega, \quad p_o = 0 \quad on \quad \Gamma, \\ \alpha\mathbf{u}_o - \nu\Delta\mathbf{u}_o = \mathbf{f} - \nabla p_o \quad in \quad \Omega, \quad \mathbf{u}_o = \mathbf{g} \quad on \quad \Gamma, \\ -\Delta\psi_o = \nabla.\mathbf{u}_o \quad in \quad \Omega, \quad \psi_o = 0 \quad on \quad \Gamma, \end{aligned} \tag{2.38}$$

$$A\lambda = \frac{\partial\psi_o}{\partial n}|_\Gamma, \tag{2.39}$$

$$\begin{aligned} \Delta p = \nabla.\mathbf{f} \quad in \quad \Omega, \quad p = \lambda \quad on \quad \Gamma, \\ \alpha\mathbf{u} - \nu\Delta\mathbf{u} = \mathbf{f} - \nabla p \quad in \quad \Omega, \quad \mathbf{u} = \mathbf{g} \quad on \quad \Gamma. \end{aligned} \tag{2.40}$$

Problems (2.38), (2.40) are classical Dirichlet problems for which quite efficient solvers exist.

Solving (2.39) is in principle more difficult since the (pseudo-differential) operator A is not known explicitely in general ; indeed we should find in [10], [12] several iterative and direct methods for solving the Stokes problem via the solution of (2.39). In this paper, we complete the analysis done in the two above references by using Fourier Analysis to construct a preconditioning operator which, according to our numerical experiments, yields conjugate gradient algorithms with much better convergence properties than those discussed in [12].

2.5 The fully implicit optimal control algorithm.

Using splitting operators for time integration, to be described in Sect. 5.2, we consider now the problem of the non linear terms of the Navier-Stokes

equation, which leads, at each full step, to solve the following non linear elliptic system [24] :

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + (\mathbf{u}.\nabla)\mathbf{u} = \mathbf{f} \quad in \quad \Omega, \mathbf{u} = \mathbf{g} \quad on \quad \gamma. \tag{2.41}$$

We introduce now the following functional spaces of Sobolev's type

$$V_o = (H_0^1(\Omega))^3,$$

$$V_g = \{\mathbf{v} | \mathbf{v} \in (H^1(\Omega))^3, \mathbf{v} = \mathbf{g} \quad on \quad \Gamma\};$$

Let $\mathbf{v} \in V_g$; from $\mathbf{v}$, we define $\mathbf{y}$ (= $\mathbf{y}$ ($\mathbf{v}$)) $\in V_o$ as the solution of

$$\alpha \mathbf{y} - \nu \Delta \mathbf{y} = \alpha \mathbf{v} - \nu \Delta \mathbf{v} + (\mathbf{v}.\nabla)\mathbf{v} - \mathbf{f} \quad in \quad \Omega, \mathbf{y} = 0 \quad on \quad \Gamma. \tag{2.42}$$

We observe that $\mathbf{y}$ is obtained from $\mathbf{v}$ via the solution of 3 uncoupled linear Poisson problems (one for each component of $\mathbf{y}$) ; it can be shown that problem (2.42) is actually equivalent to the linear variational problem

Find $\mathbf{y} \in V_o$ such that, $\forall \mathbf{z} \in V_o$ we have

$$\alpha \int_\Omega \mathbf{y}.\mathbf{z}dx + \nu \int_\Omega \nabla \mathbf{y}.\nabla \mathbf{z}dx = \alpha \int_\Omega \mathbf{v}.\mathbf{z}dx + \nu \int_\Omega \nabla \mathbf{v}.\nabla \mathbf{z}dx + \tag{2.43}$$

$$+ \int_\Omega (\mathbf{v}.\nabla)\mathbf{v}.\mathbf{z}dx - \int_\Omega \mathbf{f}.\mathbf{z}dx,$$

which has a unique solution. Suppose now that $\mathbf{v}$ is a solution of the nonlinear problem (2.41) ; the correspondingy (obtained from the solution of (2.42) (2.43)) is clearly $\mathbf{y} = 0$. From this observation, it is quite natural to introduce the following (nonlinear) least squares formulation of (2..41)

$$\text{Find } \mathbf{u} \in V_g \text{ such that } J(\mathbf{u}) \leq J(\mathbf{v}), \quad \forall \mathbf{v} \in V_g, \tag{2.44}$$

where J : $(H^1(\Omega))^3$ - > R is the function of $\mathbf{v}$ defined by

$$J(\mathbf{v}) = \frac{1}{2}\int_\Omega \{\alpha|\mathbf{y}|^2 + \nu|\nabla \mathbf{y}|^2\}dx; \tag{2.45}$$

and where y is defined fromv by (2.42), (2.43). We observe that if u is a solution of (2.41), such that J(u) = 0, then it is also a solution of (2.44). Conversely, if u is a solution of (2.44) such that J(u) = 0, then it is also a solution of (2.41).

Let be J' the derivative of J. We describe now the conjugate gradient algorithm for the previous problem :

Step 0 : Initialization

$\mathbf{u}^o \in V_g$, given,
we define then $\mathbf{g}^o, \mathbf{w}^o \in V_o$ by

$$\begin{gathered} \mathbf{g}^o \in V_o, \\ \alpha\int_\Omega \mathbf{g}^o.\mathbf{z}dx + \nu\int_\Omega \nabla \mathbf{g}^o.\nabla \mathbf{z}dx = < J'(\mathbf{u}^o), \mathbf{z} >, \forall \mathbf{z} \in V_o, \\ w^o = g^o \end{gathered} \tag{2.46}$$

with the linear functional J'($\mathbf{u}^o$) from V_o to R defined by :

$$< J'(\mathbf{u}^o), \mathbf{z} >= \alpha\int_\Omega \mathbf{y}.\mathbf{z}dx + \nu\int_\Omega \nabla \mathbf{y}.\nabla \mathbf{z}dx + \int_\Omega \mathbf{y}.(\mathbf{u}^o.\nabla)\mathbf{z}dx + \int_\Omega \mathbf{y}.(\mathbf{z}.\nabla)\mathbf{u}^o dx$$

then for $m \geq 0$, assuming that $\mathbf{u}^m, \mathbf{w}^m, \mathbf{g}^m$, are known, we obtain

$\mathbf{u}^{m+1}, \mathbf{g}^{m+1}, \mathbf{w}^{m+1}$ as follows :

Step 1 : Descent

Find $\lambda_m \in$ R such that

$$J(\mathbf{u}^m - \lambda_m \mathbf{w}^m) \leq J(\mathbf{u}^m - \lambda \mathbf{w}^m), \forall \lambda \in R, \tag{2.47}$$

$$\mathbf{u}^{m+1} = \mathbf{u}^m - \lambda_m \mathbf{w}^m.$$

Step 2 : Calculation of the new descent direction

Find $\mathbf{g}^{m+1} \in V_o$ such that

$$\alpha \int_\Omega \mathbf{g}^{m+1}.z dx + \nu \int_\Omega \nabla \mathbf{g}^{m+1}.\nabla z dx =< J'(\mathbf{u}^{m+1}), z >, \forall z \in V_o \qquad (2.48)$$

$$\gamma_m = (\alpha \int_\Omega |\mathbf{g}^{m+1}|^2 dx + \nu \int_\Omega |\nabla \mathbf{g}^{m+1}|^2 dx)/(\alpha \int_\Omega |\mathbf{g}^m|^2 dx + \nu \int_\Omega |\nabla \mathbf{g}^m|^2 dx)$$

$$\mathbf{w}^{m+1} = g^{m+1} + \gamma_m \mathbf{w}^m. \qquad (2.50)$$

Do m = m+1 and go to (2.47).

Conjugate gradient algorithms can be quite effective as solution methods for very large scale nonlinear problems [12]. Algorithm (2.46)-(2.50) is known as a Fletcher-Reeves conjugate gradient algorithm. To solve the least squares problem (2.44), it requires the solution at each iteration of exactly three Dirichlet systems (i.e. 9 scalar Dirichlet problems) associated to the elliptic operator $\alpha I - \nu\Delta$.

3. DOMAIN REDUCTION METHODS

3.1 With the stream vector

We have noted that (2.6) is an exact solution to the Navier-Stokes equation in a major part of the domain Ω. To take this into account, we consider the following decomposition :

Let $\Omega_r \subset \Omega$ be an approximation of the domain where $\nabla \times \mathbf{u} \neq 0$. If in $\Omega - \Omega_r$,

$\nabla \times \mathbf{u} = 0$ then there exists ψ such that $\mathbf{u} = \nabla \times \psi$. Instead of computing (2.23), (2.24) on Ω we compute it on Ω_r only as follows.

$$\text{Solve } \nabla \times \nabla \times \psi^{n+1} = \nabla \times \mathbf{u}^n in \Omega; \psi \times \mathbf{n}|_\Gamma = \mathbf{f} \qquad (3.1)$$

Solve (2.23)-(2.24) only in $\Omega_r with \mathbf{u}|_{\delta\Omega_r} = \nabla \times \psi^{n+1}(3.2); call \mathbf{u}^{n+1} the solution.$

In 3D (3.1) is well posed only if ψ^{n+1} is divergence free (see [13]). Figure 3.1.1 shows a 2D flow computed by this method.

3.2 With the potential

A similar decomposition can be done with the potential:
Let φ be the solution of (2.7) and let $\mathbf{v}$ be defined by

$$\mathbf{v} = \mathbf{u} - \nabla\varphi \tag{3.3}$$

Then $\mathbf{v}$ is the solution of

$$\mathbf{v}_{,t} + (\mathbf{v} + \nabla\varphi)\nabla\mathbf{v} + \nabla(p + \varphi_t + \frac{|\nabla\varphi|^2}{2}) = -\nu\nabla(\nabla\varphi); \nabla.\mathbf{v} = 0. \tag{3.4}$$

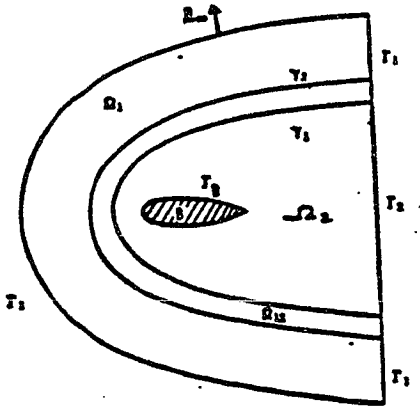
$$\mathbf{v}|_\Gamma = \mathbf{g} - \nabla\varphi \tag{3.5}$$

Figure 3.2.1 shows the result of the simulation of (3.4)-(3.5) when (2.7) is integrated on Ω while (3.4) is integrated only in $\Omega_r \subset \Omega$ and on the external boundary of Ω_r the value of $\mathbf{v}$ is taken to be 0.

3.3 With matching

Using the notation of Figure 3.3.1, let us consider the flow of an incompressible viscous fluid around the obstacle B ; we denote by ∂B the boundary of B and by Ω the flow domain, i.e. $\Omega : R^N - B$. Such a flow is modelled by the unsteady Navier-Stokes equations (with classical notation) (2.1)-(2.5).

Figure 3.3.1



In view of the numerical solution, we introduce the following simplifications : (a) we replace Ω by a sufficiently large bounded computational domain -still denoted by Ω- whose external boundary will be denoted by $\Gamma\infty$; (b) we suppose that the viscosity ν is sufficiently small so that viscous effects are

concentrated close to B and in its wake. We take advantage of these simplifications by splitting Ω into the two overlapping subdomains Ω_1 and Ω_2 -with Ω_2 containing B- and by assuming that the flow is inviscid potential in Ω_1, and still modelled by (2.1), (2.5) in Ω_2. The boundary conditions on Γ_∞ will be the following :

$$\mathbf{u} = \mathbf{u}_\infty \quad on \quad \partial\Omega_2 \cap \Gamma_\infty; \quad and \quad \frac{\partial \phi}{\partial n} = \mathbf{u}_\infty.\mathbf{n} \quad on \quad \partial\Omega_1 \cap \Gamma_\infty \tag{3.6}$$

where ϕ is the flow potential, such that $\nabla\phi = \mathbf{u}$ in Ω_1 and $\mathbf{n}$ is the unit outward normal vector. Finally, we define $\Omega_{12} = \Omega_1 \cap \Omega_2$ and γ_1 ($resp\gamma_2$) as the interface of Ω_{12} and Ω_2 ($resp\Omega_1$), (see Figure 3.3.1).

From a mathematical point of view, the coupled problem can be formulated as follows :

Find a triple $\{\phi, \mathbf{u}, p\}$ such that for each time t, we have

$$\begin{gathered} \mathbf{u}_{,t} - \nu\Delta\mathbf{u} + (\mathbf{u}.\nabla)\mathbf{u} + \nabla p = 0 \quad in \quad \Omega_2, \quad \nabla.\mathbf{u} = 0 \quad in \quad \Omega_2, \\ \mathbf{u}(x,0) = \mathbf{u}_o(x) \quad in \quad \Omega_2, \\ \mathbf{u} = 0 \quad on \quad \partial B, \quad \mathbf{u} = \mathbf{u}_\infty \quad on \quad \partial\Omega_2 \cap \Gamma_\infty, \end{gathered} \tag{3.7}$$

$$\Delta\phi = 0 \quad in \quad \Omega_1, \quad \frac{\partial \phi}{\partial n} = \mathbf{u}_\infty.\mathbf{n} \quad on \quad \partial\Omega_1 \cap \Gamma_\infty; \tag{3.8}$$

$$\int_{\Omega 12} |\nabla\phi - \mathbf{u}|^2 dx \text{ is minimal.} \tag{3.9}$$

Remark :

The above least squares methods for matching viscous and potential solutions have been introduced in the context of time dependent problems ; in fact, it can also be used to capture steady state solutions by integrating on a large time interval.

The time discretization of the Navier-Stokes equations by operator splitting methods (see 5.3) can be applied to the solution of the matching problem ; in fact, we can take advantage of the operator splitting to require the optimal matching for the solutions of linear subproblems defined on a sequence of

discrete times. The time discretization will be done through the Peaceman-Rachford operator splitting scheme.

Description of the discrete time scheme :

For Δt (> 0) a time discretization step we have

$$\mathbf{u}^o = \mathbf{u}_o \quad in \quad \Omega_2;$$

for $n \geq 0, \mathbf{u}^n$ being known on Ω_2, we look for $\{\phi^{n+1/2}, \mathbf{u}^{n+1/2}, p^{n+1/2}\}$ solution of

$$\frac{(\mathbf{u}^{n+\frac{1}{2}} - \mathbf{u}^n)}{\Delta t/2} - \frac{\nu}{2}\Delta \mathbf{u}^{n+\frac{1}{2}} + \nabla p^{n+\frac{1}{2}} = \frac{\nu}{2}\Delta \mathbf{u}^n - (\mathbf{u}^n.\nabla)\mathbf{u}^n \quad in \quad \Omega_2,$$

$$\nabla.\mathbf{u}^{n+\frac{1}{2}} = 0 \quad in \quad \Omega_2, \tag{3.10}$$

$$\mathbf{u}^{n+\frac{1}{2}} = 0 \quad on \quad \partial B, \quad \mathbf{u}^{n+\frac{1}{2}} = \mathbf{u}_\infty^{n+\frac{1}{2}} \quad on \quad \partial\Omega_2 \cap \Gamma_\infty,$$

$$\Delta\phi^{n+\frac{1}{2}} = 0 \quad in \quad \Omega_1, \frac{\partial\phi^{n+1/2}}{\partial n} = \mathbf{u}_\infty^{n+\frac{1}{2}}.\mathbf{n} \quad on \quad \partial\Omega_1 \cap \Gamma_\infty, \tag{3.11}$$

$\int_{\Omega 12} |\nabla\phi^{n+\frac{1}{2}} - \mathbf{u}^{n+\frac{1}{2}}|^2 dx$ is minimal, and then $\mathbf{u}^{n+1}$ solution of (3.12)

$$\frac{(\mathbf{u}^{n+1} - \mathbf{u}^{n+1/2})}{\Delta t/2} - \frac{\nu}{2}\Delta \mathbf{u}^{n+1} + (\mathbf{u}^{n+1}.\nabla)\mathbf{u}^{n+1} = \frac{\nu}{2}\Delta \mathbf{u}^{n+\frac{1}{2}} - \nabla p^{n+\frac{1}{2}}$$

$$in \quad \Omega_2, \tag{3.13}$$

$$\mathbf{u}^{n+1} = 0 \quad on \quad \partial B, \mathbf{u}^{n+1} = \mathbf{u}_\infty^{n+1} \quad on \quad \partial\Omega_2 \cap \Gamma_\infty, \mathbf{u}^{n+\frac{1}{2}} \quad on \quad \gamma_2$$

We shall therefore concentrate on the solution of the subproblems (3.10)-(3.12). Omitting the superscripts in (3.10), this last problem is clearly a particular case of

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = f on \Omega_2, \quad \nabla . \mathbf{u} = 0 \qquad in \quad \Omega_2, \tag{3.14}$$
$$\mathbf{u} = \mathbf{g}_2 \quad on \quad \partial B \cup (\partial \Omega_2 \cap \Gamma_\infty),$$

$$\Delta \phi = 0 \quad in \quad \Omega_1, \quad \partial \phi / \partial \mathbf{n} = \mathbf{g}_1 \qquad on \quad \partial \Omega_1 \cap \Gamma_\infty, \tag{3.15}$$

$$\int_{\Omega 12} |\nabla \phi - \mathbf{u}|^2 dx \text{ is minimal.} \tag{3.16}$$

In view of solving the above matching problem (3.14), (3.16), it is very convenient to take as master variables the trace of ϕ on γ_1 and the trace of u on γ_2. Once these traces have been specified ϕ, u and p are uniquely defined (p is in fact defined within an arbitrary additive constant and the trace z of u on γ_2 has to satisfy :

$$\int_{\gamma 2} \mathbf{z}.\mathbf{n} d\gamma_2 + \int_{\Gamma \infty \cap \partial \Omega 2} \mathbf{g}_2.\mathbf{n} d\Gamma_2 = 0). \tag{3.17}$$

Problems (3.14), (3.16) has in fact the structure of an optimal control problem whose solution will be briefly discussed in the following lines.

We formulate problems (3.14), (3.16) as an optimal control problem by

$$\min_{\{\eta, \mathbf{z}\} \in V_1 \times V_2} J(\eta, \mathbf{z}); \tag{3.18}$$

where V_1 is a space of suitable functions defined over γ_1, and

$$V_2 = \{\mathbf{z} | \mathbf{z} \text{ satisfies } \int_{\gamma 2} \mathbf{z}.\mathbf{n} d\gamma_2 + \int_{\Gamma \infty \cap \partial \Omega 2} \mathbf{g}_2.\mathbf{n} d\Gamma_2 = 0\}$$

with

$$J(\eta, \mathbf{z}) = \frac{1}{2} \int_{\Omega 12} |\mathbf{u} - \nabla \phi|^2 dx, \tag{3.19}$$

where ϕ and u are solutions of the following problems

$$\Delta\phi = 0 \quad in \quad \Omega_1, \quad \frac{\partial\phi}{\partial n} = \mathbf{u}_\infty.\mathbf{n} \quad on \quad \gamma_\infty \cap \partial\Omega_1, \quad \phi = \eta \quad on \quad \gamma_1, \quad and \tag{3.20}$$

$$\alpha\mathbf{u} - \nu\Delta\mathbf{u} + \nabla p = \mathbf{f} \quad in \quad \Omega_2, \quad \nabla.\mathbf{u} = 0 \quad in \quad \Omega_2, \tag{3.21}$$

$$\mathbf{u} = \mathbf{g}_2 \quad on \quad \partial B \cup (\partial\Omega_2 \cap \Gamma_\infty), \quad \mathbf{u} = \mathbf{z} \quad on \quad \gamma_2; \tag{3.22}$$

ϕ and $\mathbf{u}$ appears therefore as the solutions of a Poisson problem and of a Stokes problem, respectively. Efficient solvers exist now for such problems.

Using variational principles and finite element approximations like those discussed in e.g. [14], we can reduce problem (3.14), (3.16), (3.18) to a finite dimensional linear quadratic control problem, which can be solved by a conjugate gradient algorithm. Such an algorithm requires at each iteration the solution of two discrete Poisson problems on Ω_1, and of two discrete Stokes problems on Ω_2.

The lengthy technical details associated with the above procedure are fully describes in [15]. Let's mention, however, that a key step is to obtain the adjoint (co-state) equations associated to (3.20), (3.21), (3.22) ; this is done through classical arguments of Control Theory for Partial Differential Equations..

4. BOUNDARY CONDITIONS

There are several reasons to work with more general boundary conditions than the no-slip conditions:

- Absorbing outflow boundary conditions
- Turbulent slip boundary conditions or kurten boundary layer for rarefied gas dynamics
- Approximation of infinity by more sophisticated boundary conditions.

4.1. The turbulence modelling of the boundary layer

Provided that the solution stays smooth the previous code still gives meaningful results when ν tends to zero. However in pratice two things can occur :

1. As $\nu \rightarrow 0$ the results don't seem to depend on ν
2. As $\nu \rightarrow 0$ local oscillations occur.

Case 1 tends to appear for stationnary flows ; it gives to the user the illusion that high Reynolds number flows are computable while he really solves

Euler equations. Case 2 arise usually for transient flows because the numerical viscosity is not sufficient ; there is no contradiction with the theoretical error estimates because here, unlike case 1, $\|\mathbf{u}\|_2$, tends to infinity as ν tends to zero. Then, the only cure is to increase the number of nodes.

In the $k-\epsilon$ model [16] the no-slip boundary condition is relaxed and a matching with the logarithmic sublayer is attempted instead

$$\mathbf{u}.\mathbf{n} = 0 \tag{4.1}$$

$$\frac{\partial \mathbf{u}.\tau}{\partial n} + \alpha \mathbf{u}.\tau = -u^* \tag{4.2}$$

where u^* is computed from the value of u one grid point away from the wall.

For the aiplane problem, the Smagorinsky model [17] has been tried; it did not give any improvement to the direct simulation because the grid is not yet small enough to represent eddies in the beginning of the inertial range. Improvement may be possible however if boundary conditions like (4.1) are taken instead of (2.2). (Figure 4.1.1).

5. IMPROVEMENT OF ALGORITHMS

5.1 Preconditionings for the linear systems

We go back to the Stokes problem for getting a preconditionning which permits a good rate of convergence of the conjugate gradient method used for solving the problem (2.29). At first, we consider the case of a simple geometry in 2D which is the half plane :

$$\Omega = \{x| \quad x = (x_1, x_2); x_1 > 0, x_2 \in R\} \tag{5.1}$$

$$\Gamma = \{x | x_1 = 0, x_2 \in R\} \tag{5.2}$$

Let μ be defined over Γ ; denoting by μ the Fourier transform of μ, i.e.

$$\mu(s) = \int_{-\infty}^{+\infty} e^{-2i\pi s x_2} \mu(x_2) dx_2, \tag{5.3}$$

$$\text{then } \mu(x_2) = \int_{-\infty}^{+\infty} e^{2i\pi s x_2} \mu(s)\, ds \tag{5.4}$$

$$(A\mu)(x_2) = \int_{-\infty}^{+\infty} A(e^{2i\pi s x_2})\mu(s)\, ds. \tag{5.5}$$

From (5.3), (5.5), a possible approach to analyze A is to compute $A(e^{2i\pi s x_2})$; back to the definition of A (cf. (2.31), (2.34)) we first have to solve (concentrating on those solutions bounded at infinity) :

$$\Delta p = 0 \quad in \quad \Omega, p = e^{2i\pi s x_1} \quad on \quad \gamma. \tag{5.6}$$

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} = -\nabla p \quad in \quad \Omega, \quad \mathbf{u} = 0 \quad on \quad \gamma. \tag{5.7}$$

$$-\Delta \psi = \nabla . \mathbf{u} \quad in \quad \Omega, \psi = 0 \quad on \quad \gamma. \tag{5.8}$$

$$A(e^{2i\pi s x_2}) = -\frac{\partial \psi}{\partial n}|_\Gamma = \partial \psi / \partial x_1(0, x_2). \tag{5.9}$$

From the simplicity of Ω, we can solve the elliptic problems (5.6) to (5.8) by the method of separation of variables ; we obtain then

$$p(x_1, x_2) = e^{-2\pi|s|x_1} e^{2i\pi s x_2}, \tag{5.10}$$

$$\begin{aligned} u_1(x_1, x_2) &= 2\pi|s|/\alpha(e^{-2\pi|s|x_1} - e^{-\omega x_1})e^{2i\pi s x_2}, \\ u_2(x_1, x_2) &= 2i\pi|s|/\alpha(e^{-2\pi|s|x_1} - e^{-\omega x_1})e^{2i\pi s x_2}, \end{aligned} \tag{5.11}$$

$$\text{with } \omega = \omega(s) = (\frac{\alpha}{\nu} + 4\pi^2 s^2)^{\frac{1}{2}}), \tag{5.12}$$

$$\alpha\psi(x_1,x_2) = \frac{2\pi|s|}{(\omega+2\pi|s|)}(e^{-2\pi|s|x_1} - e^{-\omega x_1})e^{2i\pi s x_2}, \tag{5.13}$$

Combining (5.9) and (5.13), we finally obtain

$$A(e^{2i\pi s x_2}) = \frac{1}{\nu}\frac{2\pi|s|}{(\omega(s)+2\pi|s|)^2}e^{2i\pi s x_2}. \tag{5.14}$$

Let's denote by A the real valued function defined by

$$A(s) = \frac{1}{\nu}\frac{2\pi|s|}{(\omega(s)+2\pi|s|)^2}, \tag{5.15}$$

and by A its inverse Fourier transform (i.e.

$$A(x_2) = \int_{-\infty}^{+\infty} e^{2i\pi s x_2}A(s)ds).$$

It follows from (5.5); (5.14) that A is the convolution operator defined by

$$(A\mu)(x_2) = \int_{-\infty}^{+\infty} A(x_2-\xi)\mu(\xi)d\xi. \tag{5.16}$$

In order to precondition the iterative methods solving the Stokes problem (2.29) via the solution of (2.38), (2.40), we have to find an operator B, spectrally close to A^{-1} and whose synthesis is easy.

Define B by :

$$B(s) = \frac{(4\pi|s| + \sqrt{\frac{\alpha}{\nu}})^2}{2\pi|s|}; \tag{5.17}$$

we have then (if s $\neq$ 0) :

$$\nu A(s)B(s) = (\frac{(4\pi|s| + \sqrt{\frac{\alpha}{\nu}})}{(\omega(s)+2\pi|s|)})^2. \tag{5.18}$$

We should easily prove that

$$\lim_{s->0} \nu A(s)B(s) = \lim_{|s|->+\infty} \nu A(x)B(s) = 1; \tag{5.19}$$

in addition to (5.19), we have

$$\inf_{s\in R} \nu A(s)B(s) = 1, \quad \max_{s\in R} \nu A(s)B(s) = (\frac{5}{4})^2. \tag{5.20}$$

Back to (5.17), we also have

$$B(s) = \frac{\alpha}{\nu}\frac{1}{2}\pi|s| + 4\sqrt{\frac{\alpha}{\nu}} + 8\pi|s|. \tag{5.21}$$

We should easily verify that B(s) is the symbol of the pseudo-differential operator B defined by

$$B\mu = \frac{\alpha}{\nu}\phi_\mu + 4\sqrt{\frac{\alpha}{\nu}} + 4\frac{\partial\theta_\mu}{\partial \mathbf{n}}, \tag{5.22}$$

where z_μ and θ_μ are the respective solutions of the Neumann and Dirichlet problems

$$\Delta\phi_\mu = 0 \quad in \quad \Omega, \quad \frac{\partial\phi_\mu}{\partial \mathbf{n}} = \mu \quad on \quad \gamma, \tag{5.23}$$

$$\Delta\theta_\mu = 0 in \Omega \theta_\mu = \mu \quad on \quad \Gamma; \tag{5.24}$$

we assume that in (5.22)-(5.24), we have $\int_\Gamma \mu \, d\Gamma = 0$.

Verifying the symmetry of B is quite easy. The "function" $B\mu$ can be computed via the solution of simple elliptic problems ; furthermore, it follows from (5.20) that we can expect B to have good properties as a preconditioning operator for solving problem (2.29), via the solution of (2.38), (2.40).

Now we describe the modified conjugate gradient algorithm :

Step 0 : Initialization

$\lambda^o \in H^{-1/2}(\Gamma)$ is given ;
Solve then

$$\Delta p^o = \nabla . \underline{f} \quad in \quad \Omega, p^o = \lambda^o \quad on \quad \gamma, \tag{5.25}$$

$$\alpha \mathbf{u}^o - \nu \Delta \mathbf{u}^o = \mathbf{f} - \nabla p^o \quad in \quad \Omega, \quad \mathbf{u}^o = \mathbf{g} \quad on \quad \gamma, \tag{5.26}$$

$$-\Delta \psi^o = \nabla . \mathbf{u}^o \quad in \quad \Omega, \quad \psi^o = 0 \quad on \quad \gamma. \tag{5.27}$$

and define

$$r^o = -\frac{\partial \psi^o}{\partial n}|_\Gamma. \tag{5.28}$$

The preconditioning is achieved through the solution of

$$\Delta \phi^o = 0 \quad in \quad \Omega, \quad \frac{\partial \phi^o}{\partial n} = r^o on \Gamma, \quad \int_\Gamma \phi^o d\Gamma = 0 \tag{5.29}$$

$$\Delta \theta^o = 0 \quad in \quad \Omega, \quad \theta^o = r^o \quad on \quad \gamma. \tag{5.30}$$

and we set

$$g^o = \frac{\alpha}{\nu}\phi^o|_\Gamma + 4\sqrt{\frac{\alpha}{\nu}} r^o + 4\frac{\partial \theta^o}{\partial n}|_\Gamma, \tag{5.31}$$

$$w^o = g^o. \tag{5.32}$$

Then, for $n \geq 0$, with $\lambda^n, p^n, \mathbf{u}^n, \psi^n, r^n, g^n, w^n$ known, we compute $\lambda^{n+1}, p^{n+1}, \mathbf{u}^{n+1}, \psi^{n+1}, r^{n+1}, g^{n+1}, w^{n+1}$ as follows :

Step 1 : Descent

$$\Delta p^n = 0 \quad in \quad \Omega, \quad p^n = w^n \quad on \quad \gamma, \tag{5.33}$$

$$\alpha u^n - \nu \Delta u^n = -\nabla p^n \quad in \quad \Omega, \quad u^n = 0 \quad on \quad \gamma, \tag{5.34}$$

$$-\Delta \psi^n = \nabla . u^n \quad in \quad \Omega, \quad \psi^n = 0 \quad on \quad \gamma. \tag{5.35}$$

$$\rho_n = -\int_\Gamma r^n g^n d\Gamma / \int_\Gamma \partial\psi^n/\partial \mathbf{n} w^n d\Gamma = -\int_\Gamma r^n w^n d\Gamma / \int_\Gamma \partial\psi^n/\partial \mathbf{n} w^n d\Gamma. \tag{5.36}$$

We define then $\lambda^{n+1}, p^{n+1}, \mathbf{u}^{n+1}, \psi^{n+1}, g^{n+1}$, and r^{n+1} by

$$\lambda^{n+1} = \lambda^n - \rho_n w^n, \tag{5.37}$$

$$p^{n+1} = p^n - \rho_n p^n, \tag{5.38}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \rho_n u^n, \tag{5.39}$$

$$\psi^{n+1} = \psi^n - \rho_n \psi^n, \tag{5.40}$$

$$r^{n+1} = r^n + \rho_n \frac{\partial \psi}{\partial n}. \tag{5.41}$$

Step 2 : Construction of the new descent direction

Solve

$$\Delta\phi^{n+1} = 0 \quad in \quad \Omega, \quad \frac{\partial \phi^{n+1}}{\partial n} = r^{n+1} on \Gamma, \quad \int_\Gamma \phi^{n+1} d\Gamma = 0 \tag{5.42}$$

$$\Delta\theta^{n+1} = 0 \quad in \quad \Omega, \quad \theta^{n+1} = r^{n+1} on \Gamma. \tag{5.43}$$

and define g^{n+1} by

$$g^{n+1} = \frac{\alpha}{\nu}\phi^{n+1}|_\Gamma + 4\sqrt{\frac{\alpha}{\nu}} r^{n+1} + 4\frac{\partial \theta^{n+1}}{\partial n}|_\Gamma. \tag{5.44}$$

Compute then

$$\gamma_n = \frac{\int_\Gamma r^{n+1} g^{n+1} d\Gamma}{\int_\Gamma r^n g^n d\Gamma} \tag{5.45}$$

$$w^{n+1} = g^{n+1} + \gamma_n w^n. \tag{5.46}$$

Do n = n+1 and go to (5.33).

We can expect that if Ω is an half-space, the contraction ratio in (5.10) has 1/9 as upper bound, i.e. the error practically is reduced by an order of magnitude per iteration. According to the numerical experiments (figure 5.1.3) the above theoretical prediction almost holds (we obtain an order of magnitude per two iterations) despite the fact that the corresponding geometry is much more complicated (2D nozzle). We have also compared algorithm (5.25)-(5.46) to the simpler variant obtained by taking S equal to the identity operator of $L^2(\Gamma)$; the new algorithm (i.e. (5.25)-(5.46) is much faster in nomber of iterations and C.P.U. time.

5.2 θ – *Schemes*

For more details, concerning the following section, we refer to [10].

Let us consider the real Hilbert space H ; we consider then in H the following initial value problem

$$\frac{du}{dt} + A(u) = f, \quad u(0) = u_o, \tag{5.47}$$

where A is an operator from H and where f is a source term and u_o the initial value, respectively. Let A_1 and A_2 be two operators such that

$$A = A_1 + A_2. \tag{5.48}$$

With Δt (> 0) a time discretization step, let us define a scheme for solving (5.47), taking advantage of the decomposition (5.48) :

Let θ belong to the open interval]0,1/2[. The idea behind the scheme is to split the time interval $[n\Delta t, (n+1)\Delta t]$ into three subintervals, as shown in Figure 5.3.1, and integrate (approximately) with respect to time (using an implicit scheme for A_1 and an explicit scheme for A_2) on $[n\Delta t, (n+\theta)\Delta t]$, then switch the role of A_1 and A_2 on $[(n+\theta)\Delta t, (n+1-\theta)\Delta t]$ and finally, on $[(n+1-\theta)\Delta t, (n+1)\Delta t]$ do like on $[n\Delta t, (n+\theta)\Delta t]$.

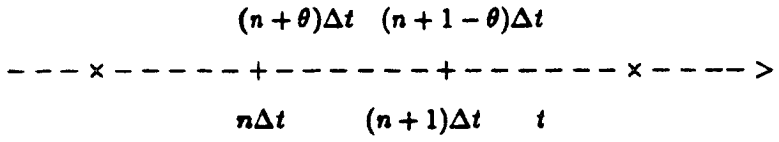


Figure 5.3.1

Using these principles, we obtain the following scheme :

$$u^o = u_o; \tag{5.49}$$

then, for n $\geq$ 0, we obtain $u^{n+\theta}, u^{n+1-\theta}, u^{n+1}$, from u^n, as follows

$$\frac{(u^{n+\theta} - u^n)}{\theta \Delta t} + A_1(u^{n+\theta}) + A_2(u^n) = f^{n+\theta}, \tag{5.50}$$

$$\frac{(u^{n+1-\theta} - u^{n+\theta})}{(1-2\theta)\Delta t} + A_1(u^{n+\theta}) + A_2(u^{n+1-\theta}) = f^{n+1-\theta}, \tag{5.51}$$

$$\frac{(u^{n+1} - u^{n+1-\theta})}{\theta \Delta t} + A_1(u^{n+1}) + A_2(u^{n+1-\theta}) = f^{n+1} \tag{5.52}$$

we don't know general results for the convergence of (5.49)-(5.52) (but, under quite general monotonicity assumption on A_1 and A_2, very likely, the methods in [18] still apply).

If we apply this scheme to the Navier-Stokes operator, we have the following scheme :

$$H = (H^1(\Omega))^3$$

$$\mathbf{u}^o = \mathbf{u}_o \in H$$

then for n $\geq$ 0, starting from $\mathbf{u}^n$ we solve

$$\frac{(\mathbf{u}^{n+\theta}-\mathbf{u}^{n})}{\theta\Delta t}-\alpha\nu\Delta\mathbf{u}^{n+\theta}+\nabla p^{n+\theta}$$
$$=\mathbf{f}^{n+\theta}+\beta\nu\Delta\mathbf{u}^{n}-(\mathbf{u}^{n}.\nabla)\mathbf{u}^{n}\quad in\quad\Omega,$$
$$\nabla.\mathbf{u}^{n+\theta}=0\quad in\quad\Omega,\quad\mathbf{u}^{n+\theta}=\mathbf{g}^{n+\theta}\quad on\quad\gamma,\tag{5.53}$$

$$\frac{(\mathbf{u}^{n+1-\theta}-\mathbf{u}^{n+\theta})}{(1-2\theta)\Delta t}-\beta\nu\Delta\mathbf{u}^{n+1-\theta}+(\mathbf{u}^{n+1-\theta}.\nabla)\mathbf{u}^{n+1-\theta}=$$
$$\mathbf{f}^{n+1-\theta}+\alpha\nu\Delta\mathbf{u}^{n+\theta}-\nabla p^{n+\theta}\dot{u}\quad in\quad\Omega,$$
$$\mathbf{u}^{n+1-\theta}=\mathbf{g}^{n+1-\theta}\quad on\quad\gamma,\tag{5.54}$$

$$(\mathbf{u}^{n+1}-\frac{\mathbf{u}^{n+1-\theta})}{\theta\Delta\mathbf{t}}-\alpha\nu\Delta\mathbf{u}^{n+1}+\nabla p^{n+1}=$$
$$\mathbf{f}^{n+1}+\beta\nu\Delta\mathbf{u}^{n+1-\theta}-(\mathbf{u}^{n+1-\theta}.\nabla)\mathbf{u}^{n+1-\theta}\quad in\quad\Omega,$$

$$\nabla.\mathbf{u}^{n+1}=0\quad in\quad\Omega,\quad\mathbf{u}^{n+1}=\mathbf{g}^{n+1}\quad on\quad\gamma.\tag{5.55}$$

$$\alpha=\frac{(1-2\theta)}{(1-\theta)};\quad\beta=\theta/(1-\theta);\quad\alpha+\beta=1.\tag{5.56}$$

Using the above operator splitting method, we have been able to decouple nonlinearity and incompressibility in the Navier-Stokes equations. We observe that $\mathbf{u}^{n+\theta}$ and $\mathbf{u}^{n+1}$ are obtained from the solutions of linear problems very close to the steady Stokes problem.

If one uses scheme (5.53)-(5.55), the best choice for α and β is given by (5.56). With such a choice, many computer subprograms can be used for both the linear and nonlinear subproblems, resulting, therefore, in quite substantial core memory savings.

5.3 Multigrids

Using the operator splitting methods of Section 5.2, we have to solve, at each full step, linear and nonlinear subproblems (5.57) and (5.58) of the following type :

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} \quad in \quad \Omega$$

$$\nabla . \mathbf{u} = 0 \quad in \quad \Omega \quad \mathbf{u} = \mathbf{g} \quad on \quad \gamma \ (\text{with} \int_\Gamma \mathbf{g}.\mathbf{n} d\Gamma = 0) \tag{5.57}$$

$$\alpha \mathbf{u} - \nu \Delta \mathbf{u} + (\mathbf{u}.\nabla)\mathbf{u} = \mathbf{f} \quad in \quad \Omega, \quad \mathbf{u} = g on \Gamma \tag{5.58}$$

This section describes how (5.57) and (5.58) can be solved by quite efficient multigrid solvers.

5.3.1. A multigrid algorithm for the Stokes subproblem (5.57)

One of the most simple ways to discretize (5.57) is to apply finite element techniques using the mini element [19] P1 buble/P1 described in Section 2 which satisfies the so-called LBB[4][5][19] stability condition.

Let $(G_h)_h$ be a family of standard nested triangulations of Ω with h the maximal length of the edges of the triangles of G_h.

We introduce the following discrete spaces (with P_k the space of the polynomials in two variables of degree $\leq$ k)

$$V_h = \{\mathbf{v}_h | \mathbf{v}_h \in C^o(\Omega) \times C^o(\Omega); \quad \mathbf{v}_h|_T \in P_{1T}^* \times P_{1T}^*, \quad \forall T \in G_h\}$$

$$Q_h = \{q_h | q_h \in C^o(\Omega); \quad q_h|_T \in P_1, \quad \forall T \in G_h\} \tag{5.59}$$

with P_{1T}^* the subspace of P_3 defined as follows :

$$P_{1T}^* = \{q | q = q_1 + \lambda \phi_T; \quad q_1 \in P_1, \quad \lambda \in R \tag{5.60}$$
$$\phi_T \in P_3, \quad \phi_T = 0 \quad on \quad \partial T, \quad \phi_T(G_T) = 1\}$$

The discrete formulation of (5.57) is then written :

Find $(\mathbf{u}_h, p_h) \in V_h \times Q_h$ such that :

$$\alpha \int_\Omega \mathbf{u}_h.\mathbf{v}_h dx + \nu \int_\Omega \nabla \mathbf{u}_h.\nabla \mathbf{v}_h dx - \int_\Omega p_h \nabla.\mathbf{v}_h dx = \int_\Omega \mathbf{f}.\mathbf{v}_h dx, \quad \forall \mathbf{v}_h \in V_{oh}. \tag{5.61a}$$

$$\int_\Omega q_h \nabla.\mathbf{u}_h dx = 0, \quad \forall q_h \in Q_h. \tag{5.61b}$$

Solving problem (5.57) requires the solution of a linear system of the following type, obtained by using static condensation technique to eliminate the buble function (cf. [20]).

$$\begin{vmatrix} A_h & B_h^t \\ B_h & C_h \end{vmatrix} \begin{vmatrix} U \\ P \end{vmatrix} = \begin{vmatrix} F_1 \\ F_2 \end{vmatrix} \tag{5.62}$$

(C_h comes from the condensation (mini element/ stabilization). Several possibilities for the value of $C_h can or iginate from other stability considerations (cf. [20])$.

The multigrid solver :

To define the (m+1)-grids algorithm, let us consider a sequence of mesh-sizes

$h_o > h_1 > ... \; h_m$ with $h_{k-1} = 2h_k$, $1 \leq k \leq m$.

For simplicity, we replace subscripts h_k by k. We define (S_k) as being (5.62) on the grid G_k.We denote by N_k the number of points of the grid G_k and define

$X_k = (R^{Nk})^2$ and $M_k = R^{Nk}$

We are interested in solving the discrete problem on the finest grid G_m .

The steps of the algorithm are the following (cf. [21]) :

- smoothing : Let $(U_m^{(0)}, P_m^{(0)}) \in X_m \times M_m$ be a given approximation to the solution of (S_m), and involving l steps of some iteration process, compute $(U_m^{(l)}, P_m^{(l)})$ and the defect

$$\begin{vmatrix} D_m \\ d_m \end{vmatrix} = \begin{vmatrix} A & B^T \\ B & C \end{vmatrix} \begin{vmatrix} U_m^{(l)} \\ P_m^{(l)} \end{vmatrix} - \begin{vmatrix} F_1 \\ F_2 \end{vmatrix}$$

- correction : compute a correction $(U_{m-1}, P_{m-1}) \in X_{m-1} \times M_{m-1}$ on the coarser grid G_{m-1} by solving the problem (S_{m-1}) with the right hand side (D_m, d_m)

The (m+1) grid step is now completed by adding the correction to $(U_m^{(l)}, P_m^{(l)})$.

Smoothing procedure

The smoothing step is the central process in multigrid algorithms, which determines the efficiency of the algorithm. We propose here a smoother based on Uzawa minimal residual solver (cf. [22]). For more details, we refer to [21].

$l > 0$ steps of the smoother, applied to the system (5.62) transform $(U^{(0)}, P^{(0)})$ into $(U^{(l)}, P^{(l)})$ as follows :

- Let U be an approximation of U obtained from one or two sweeps of the pointwise GAUSS-SEIDEL iteration on :

A_k U* = $F_1 - B_h^t\ P^{(0)}$, starting from U* = $U^{(0)}$

and set $R_o^{(0)} <-- F_2$ - (B_h U + $C_h\ P^{(0)}$)

- For k = 0 to (l -1) :

Compute an approximation Z_k of the solution of the following system :

A_h Z = $B_h^t\ R_k$, by using one sweep of the pointwise Gauss-Seidel iteration.

Set : $W_k <-- F_2$ - ($B_h Z_k$ - $C_h(P^{(k)})$

$$\rho_k <-- \frac{(W_k, R_k)}{(W_k, W_k)}$$

$$P^{(k+1)} <-- P^{(k)} - \rho_k R_k$$

$$U^{(k+1)} <-- U^{(k)} + \rho_k Z_k$$

$$R^{(k+1)} <-- R^{(k)} - \rho_k W_k$$

Figures (5.3.1)a shows the convergence rate of this multigrid algorithm and Figure (5.3.1)b the efficiency of the full multigrid variant.

5.3.2. A multigrid method for the nonlinear subproblem (5.58)

The multigrid method is used to solve the uncoupled Poisson problems which occurs after buble condensation, and which are solved with a V-cycle algorithm.

We have used a Gauss-Seidel forward and backward sweep as smoothing procedure. As triangulations are nested, respective piecewise linear function spaces are embedded, so we have used as interpolation operator the natural injection between finite element spaces. In practice, this means that a vertex Q in G_{k+1} - G_k is affected with the average of the two vertices defining the edge with middle point Q. (We call these vertices the "father vertices" of Q). As restriction operator, we have chosen the one which is naturally associated to the variational formulation . In practice, each vertex $Q \in G_k$ receives the contribution of itself, weighted by 1 and all vertices having Q as a father vertex, weighted by 1/2.

On the coarsest level, we have used the SSOR preconditioned conjugate gradient method to solve the defect equation and grids G_k have been generated by uniform refining. For more details, we refer to Munoz [23].

Figure (5.3.2) shows the influence of the grid on convergence rate using conjugate gradient (a) and multigrids (b) for solving the Poisson problems of (5.58). Other FAS multigrid algorithms for solving the non linear problem (5.58) are presently under investigation.

6. FUTURE DEVELOPMENTS

Among the challenging problems concerning viscous flow simulation, let us mention the following ones that we intend to investigate in the near future :

- Three-dimensional incompressible viscous flows at high Reynolds numbers,
- Implementation of multigrid solvers at various stages of our iterative procedure in order to speed up the calculation,
- Investigation of adaptive grid refinement,
- Generalization to compressible flows,
- Generalization to turbulent and reacting flows,
- Operator splitting and domain decomposition techniques and their implementation on vector and parallel machines,
- Investigation of the problems associated to flow visualization in three-dimensions.

7. ACKNOWLEDGEMENTS

We would like to thank J.F. Maitre and F. Musy for a valuable comments and careful guidance concerning multigrid acceleration techniques described in Section 5.3, and also Christiane Barny for typing this paper under considerable pressure to make the deadline. The support of DRET (grant 85/175) is also acknowledged.

REFERENCES.

[1] R. Glowinski, J. Periaux, O. Pironneau: An efficient preconditionning scheme for iterative solutions of PDE. Appl. Math. Modelling 1980 **4** p187-192.

[2] D. McLaughlin, G. Papanicolaou, O.Pironneau : Convection of microstructures, SIAM J. Appl. Math. 45, (1985), 780-797.

[3] J.L. Lions : Quelques méthodes de résolutions des problèmes non lineaires, Dunod, (1969).

[4] O. Ladyzhenskaya : Mathematical Theory of Viscous Incompressible Flow. Gordon and Breach, New-York (1963).

[5] V. Girault, P.A. Raviart : Navier-Stokes equations. Springer series in computational mathematics (1986).

[6] F. Thomasset : Implementation of F.E.M. for Navier-Stokes equations. Springer series in computational physics (1981).

[7] C. Bernardi, G. Raugel : A conforming F.E.M. for the time dependant Navier-Stokes equations. Report 84034, Labo. Numer. Anal., University Paris 6.

[8] O. Pironneau : on the transport diffusion algoriothm and its applications to the Navier-Stokes equations. Numer. Math. 38 (1982), pp. 309-332.

[9] S. Suli : Lagrange Galerkin methods. Internal Report, Univ. of Oxford (1985).

[10] M.O. Bristeau, R. Glowinski, J. Périaux : Numerical Methods for the Navier-Stokes equations. Application to the simulation of compressible and incompressible viscous flows, Computer Physics Reports, (to appear).

[11] J. Cahouet : Computational Fluid Dynamics. Lecture Series in Von Karman Institute, 1986-05, March 3, 7, Rhode-St-Genèse (Belgium) 1986.

[12] R. Glowinski : Numerical Methods for Nonlinear Variational Problems, Springer-Verlag, New York, 1984.

[13] F. El Dabaghi et O. Pironneau : Stream vectors in 3D aerodynamics, Numer. Math. 1986, 48, pp. 581-589.

[14] Q.V. Dinh : Simulation numérique en éléments finis d'écoulements de fluides visqueux incompressibles par une méthode de décomposition de domaines sur processeurs vectoriels, Thèse 3ème cycle, Paris VI, 1982.

[15] G. Terrasson : thesis (to appear).

[16] B.E. Launder and D.B. Spalding : Mathematical models of turbulence, Academic Press, 1972.

[17] J.S. Smagorinsky : Mon Weather Rev. 91, 99-164.

[18] P.L. Lions, B. Mercier : Splitting algorithms for the sum of two nonlinear operator, SIAM J. Num. Anal. 16 (1979), pp. 946-979.

[19] D.N. Arnold, F. Brezzi, F. Fortin : A stable finite element for the Stokes equations (Pavia 1983).

[20] F. Brezzi , J. Pitkäranta : On the Stabilization of finite element approximations of the Stokes equations. W; Hackbush (Ed.), Efficients solutions of elliptic systems. Proceedings, Kiel, Jan. 1984, (Vol. 10).

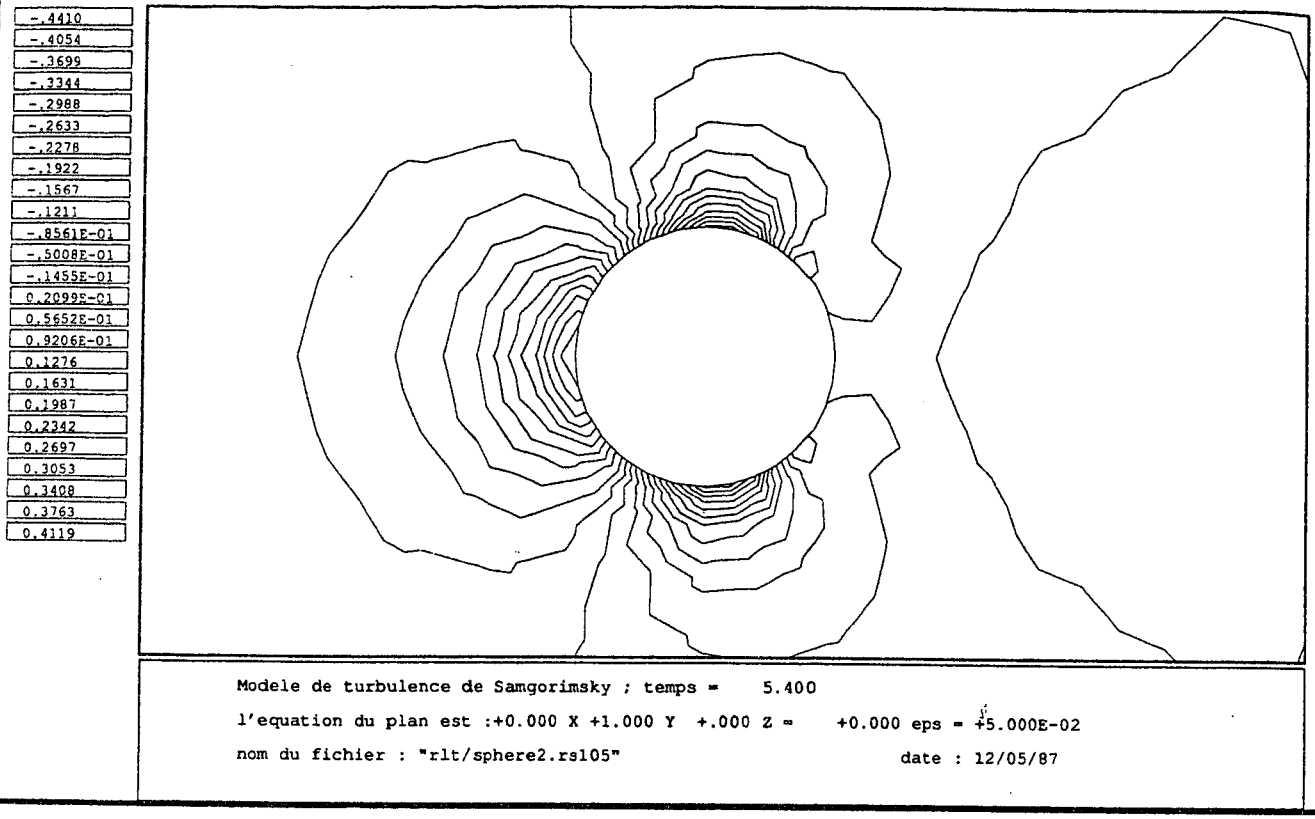
[21] M. Abdalas : thesis (to appear).

[22] J.F. Maitre : Multigrid methods for variational problems with emphasis on mixed formulations -convergence theory and numerical experiments. 2nd European Conference on Multigrid Methods - Cologne, 1985.

[23] R. Munoz : thesis (to appear).

[24] M.O. Bristeau, R. Glowinski, B. Mantel, J. Périaux, P. Perrier : Finite Elements Methods for Solving the Navier-Stokes Equations for Compressible Unsteady Flows, Proc.of 5th International Conference on Finite Element and Flow Problems, University of Texas at Austin, U.S.A., 1984, (Carey G.F. and Oden J.T. eds., pp. 449-462).

Fig. 2.2.1. Solution of the Navier-Stokes equations (3D) with Lagrangian characteristic time scheme.



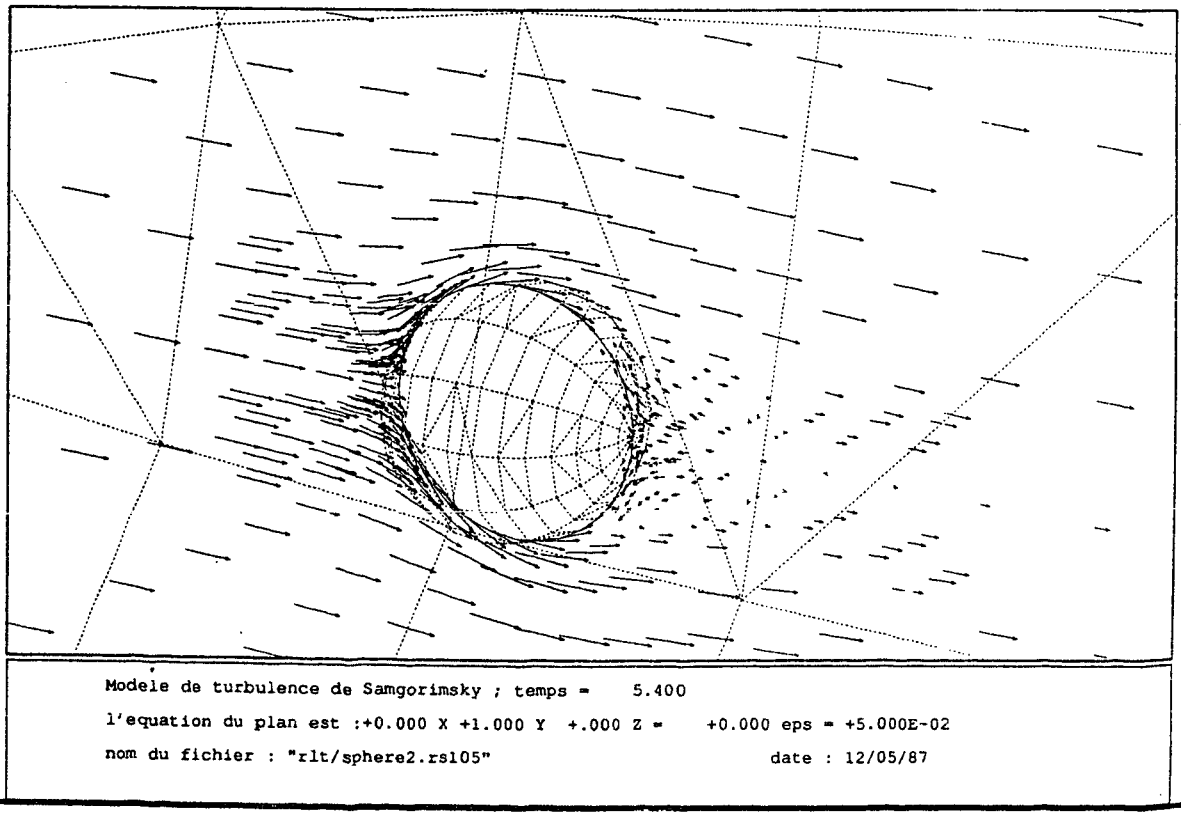
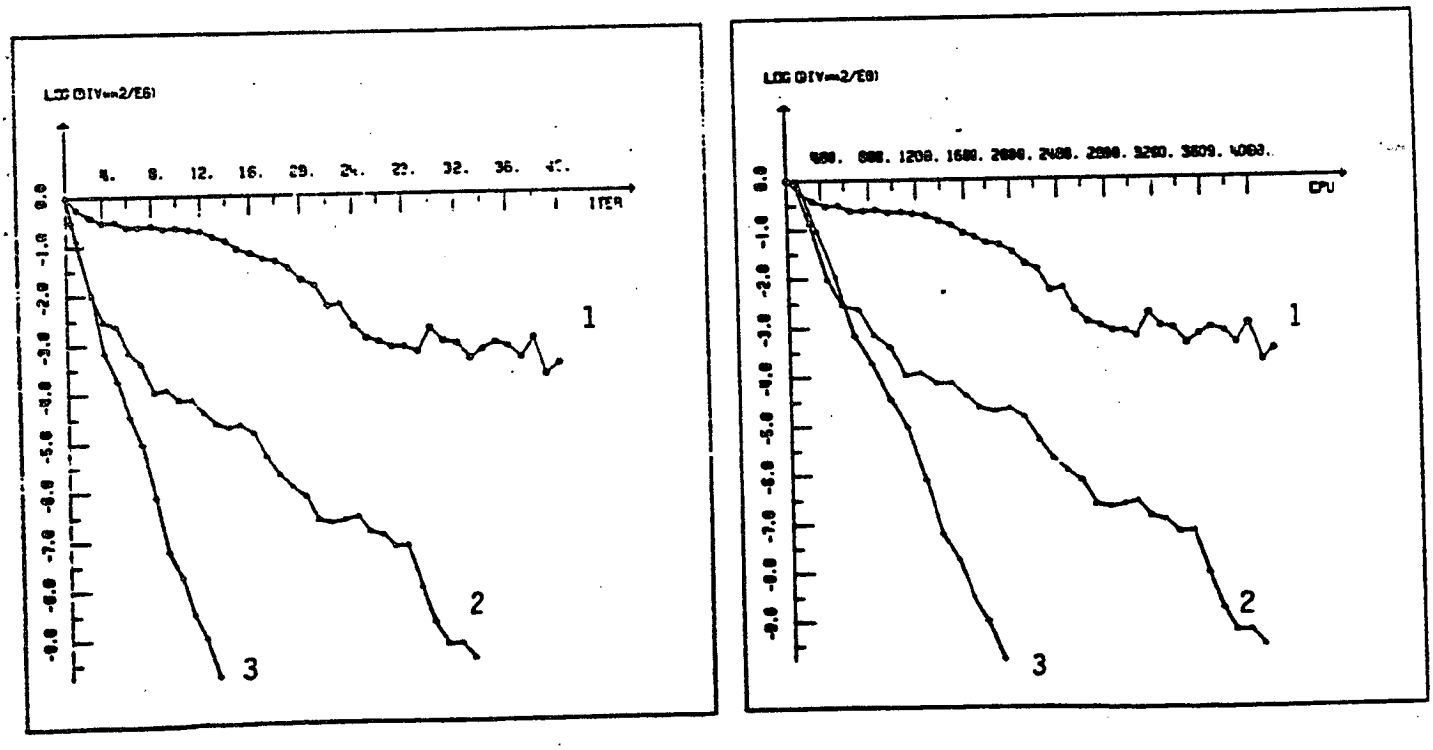
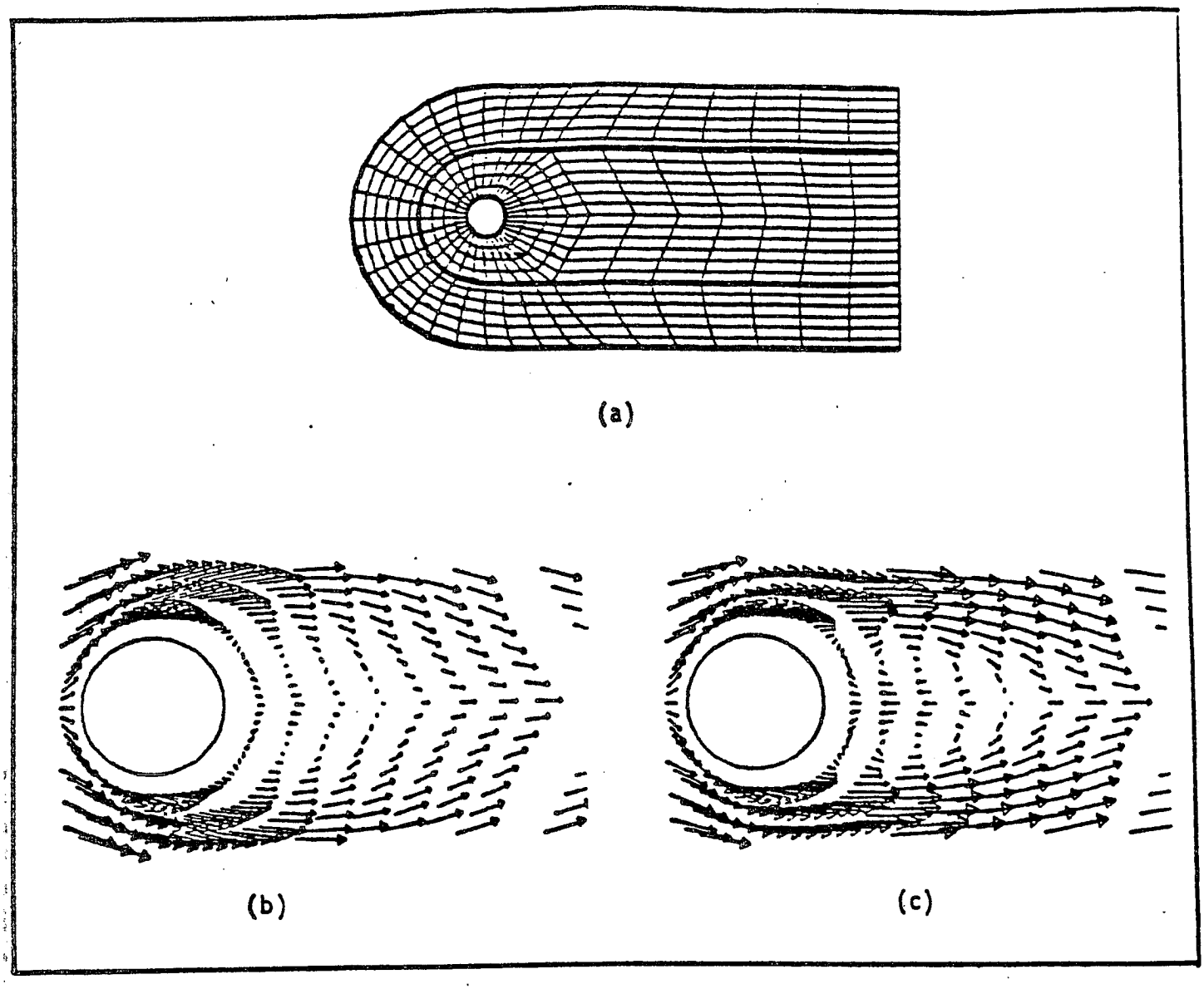


Fig. 2.3.1 - Solution of the Stokes subproblem by the Uzawa conjugate gradient algorithm



1. Preconditioning by I^{-1}
2. Preconditioning by $-\Delta^{-1}$
3. Preconditioning by $\nu I^{-1} - k\,\Delta^{-1}$

Fig. 3.1.1 - Domain reduction with ψ



Result of the algorithm : $\Omega_1 \supset \Omega$

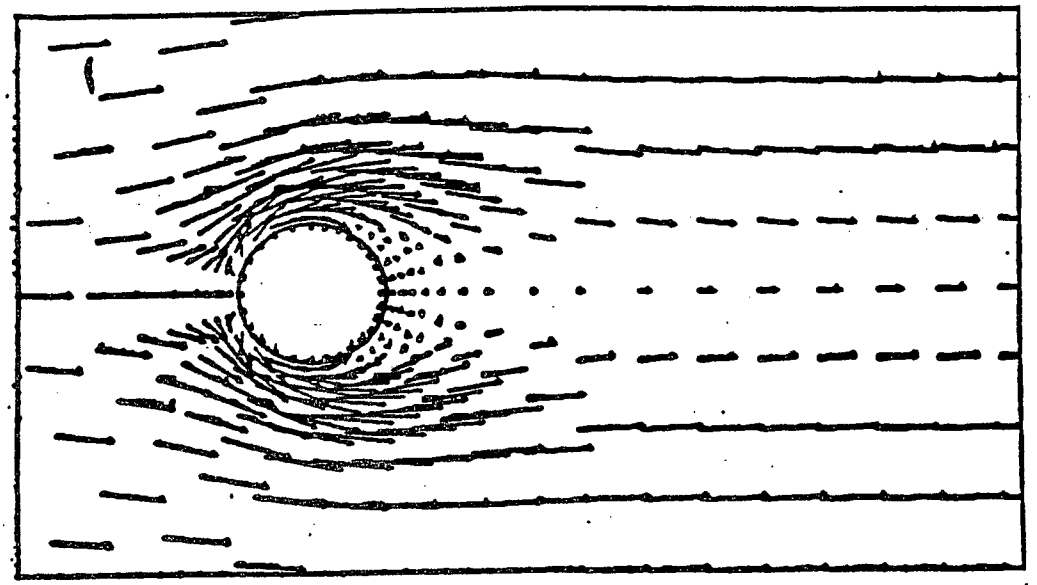
- ψ in Ω_1 / $-\Delta\psi = \omega^1$

- **u** in Ω / **u** solution of the Navier-Stokes equations with the boundary conditions $u_1 = \partial\psi/\partial x_2$
  $u_2 = -\partial\psi/\partial x_1$

- ω^1 in Ω_1 / $\omega^1 = \nabla \times \mathbf{u}$ in Ω
  $= 0$ in $\Omega_1 - \Omega$

The two domains (a), and the comparison between the result of algorithm (b) and the computation in the whole domain by the Navier-Stokes equations (c). The differences can be explained by the too small distance between the cylinder and the boundary of the small domain Ω.

Fig. 3.2.1 - Domain reduction with the potential
Re = 50. Dt = 0.8



Velocity

Vorticity

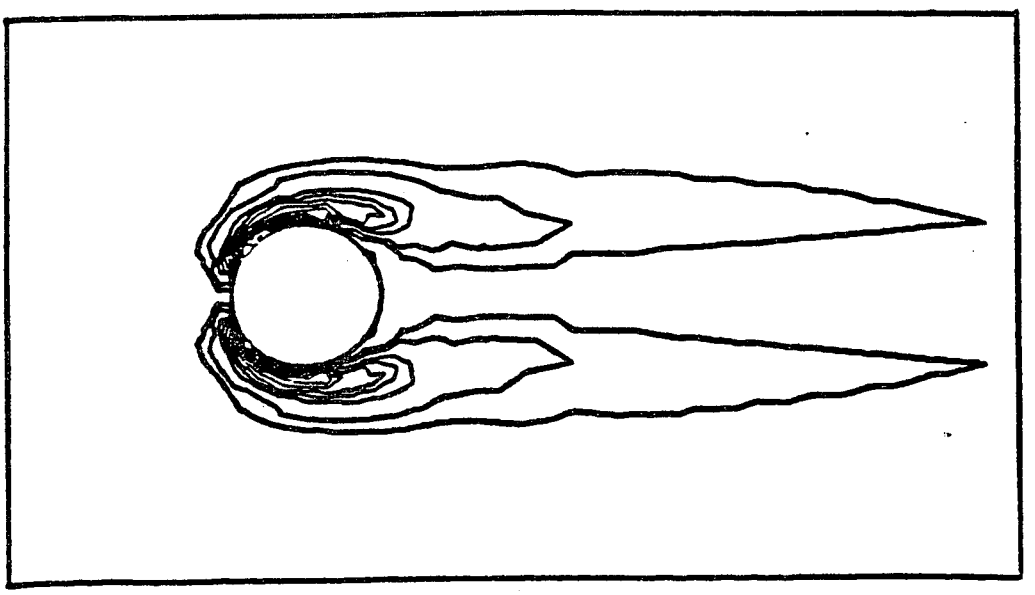
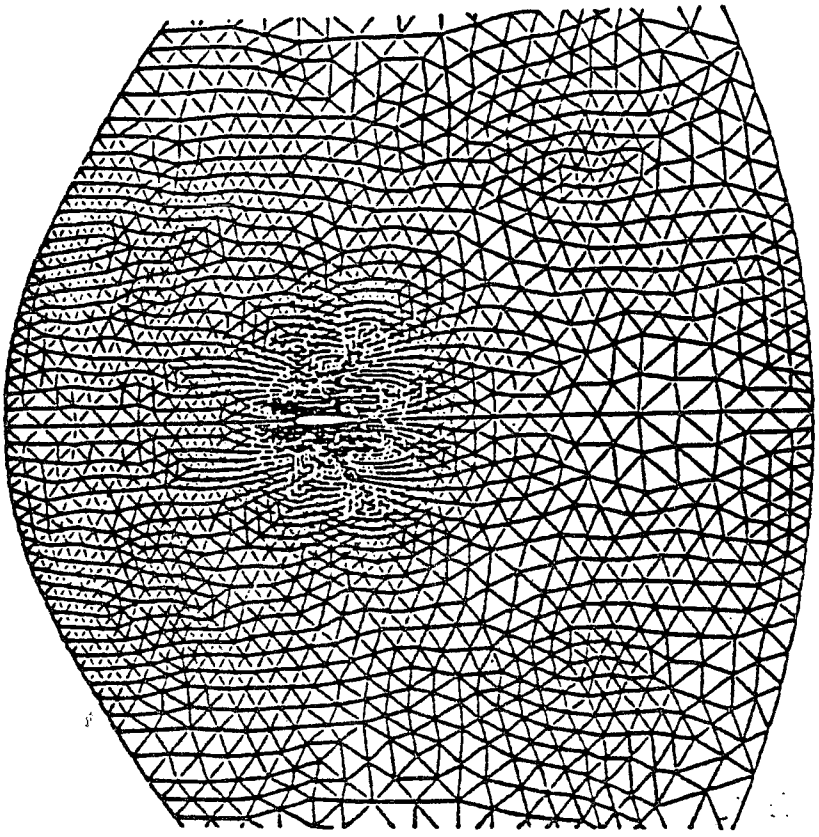
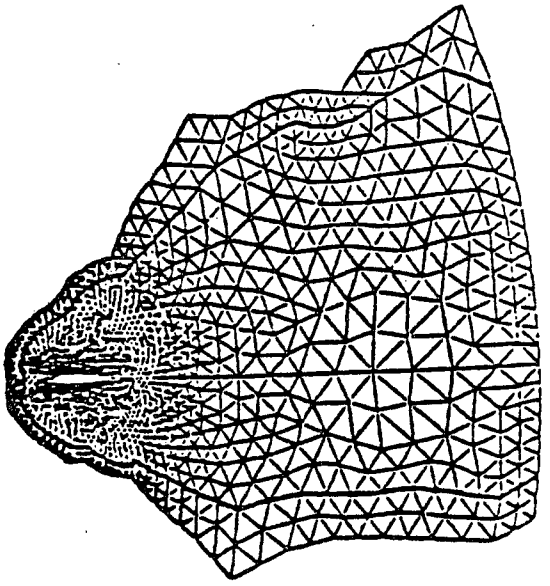


Fig. 3.3.1 - Domain reduction with matching

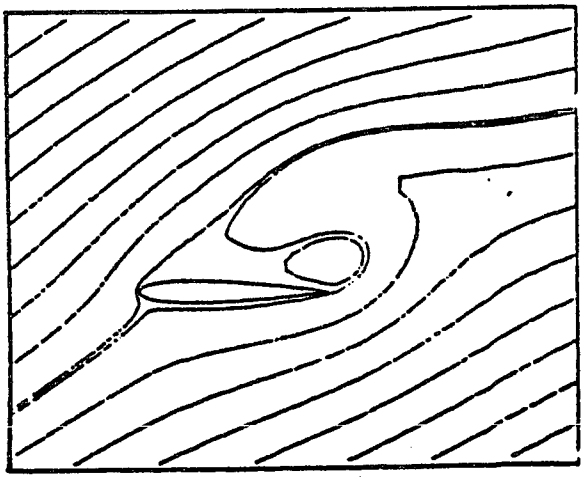
Naca0012 Re = 200. Incidence 30° Dt = 0.1



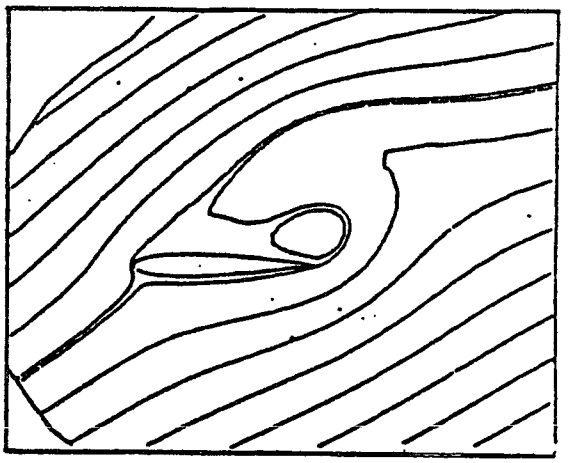
Global mesh



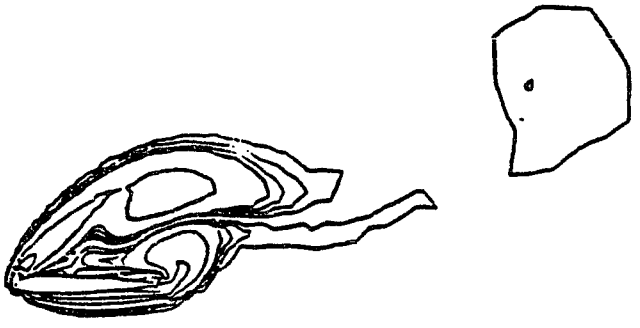
Computational mesh for viscous calculation



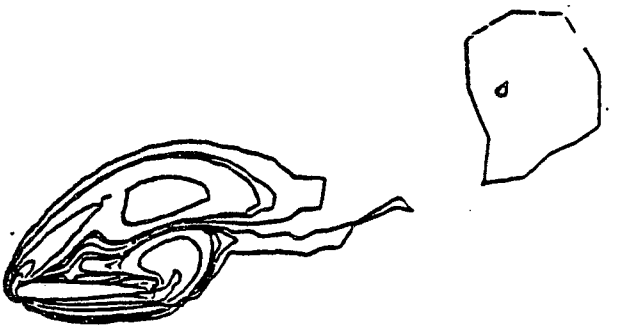
Streamlines (global calculation)



Streamlines (matching calculation)

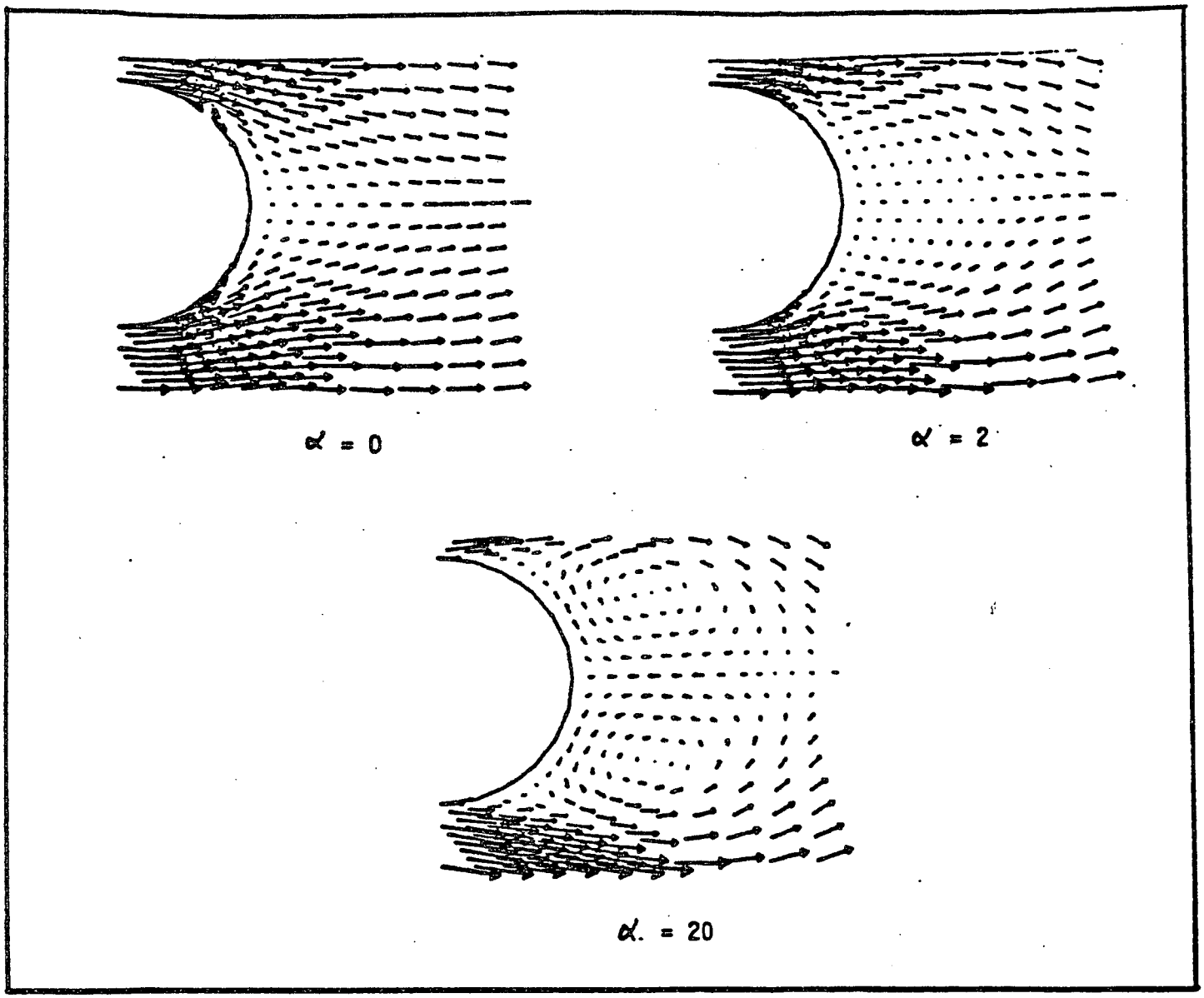


Vorticity (global calculation)



Vorticity (matching calculation)

Fig. 4.1.1 - Turbulence modelling of the boundary layer



Influence of the parameter α (rugosity of the wall) in the boundary condition :

$$\alpha\, u_\tau + \partial u_\tau / \partial n = 0 \quad \text{on } \partial S$$

$$u.n = 0$$

where S is the cylinder and u_τ the tangential component of the velocity in the Navier-Stokes equations.

Fig. 5.1.1 - Preconditioning the linear Stokes problem

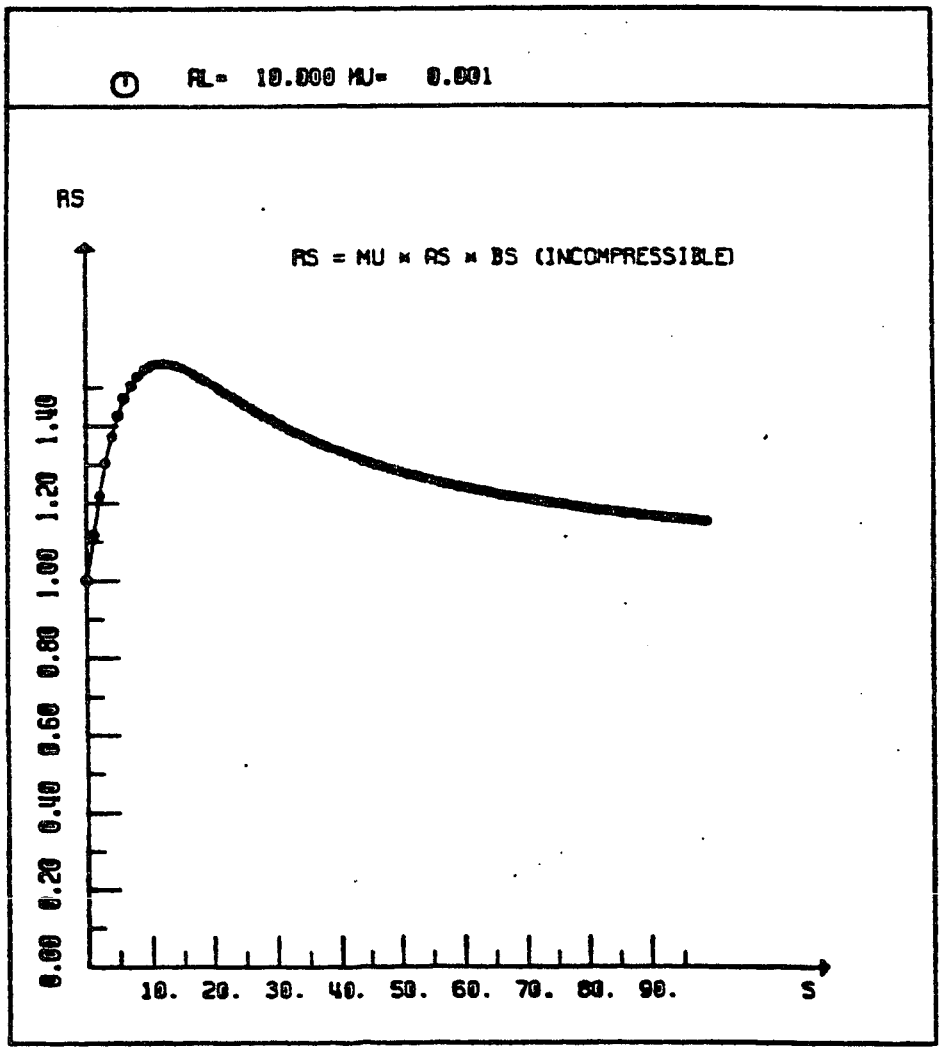


Fig. 5.1.2 - Computational grids

Fig. 5.1.2$_a$
182 boundary nodes

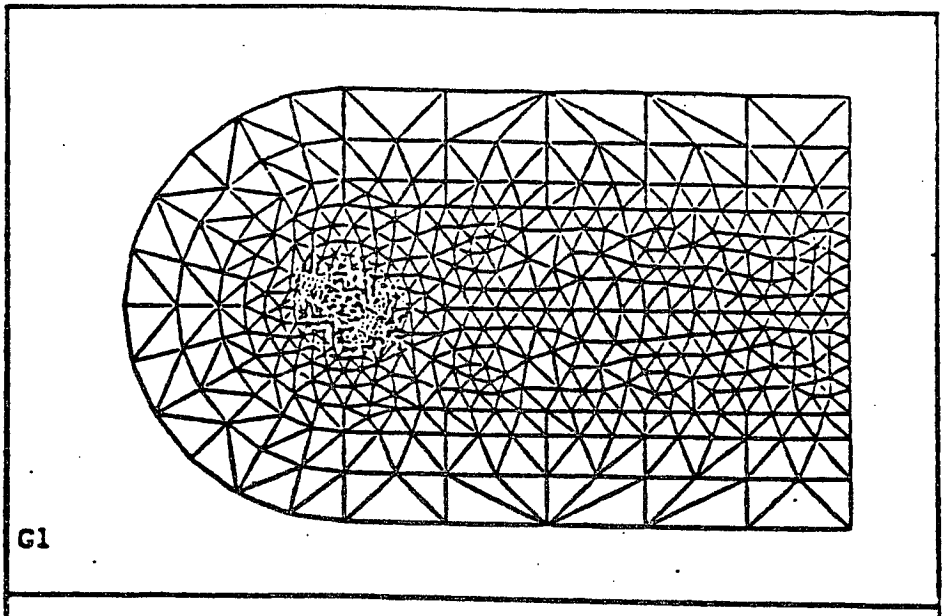


Fig. 5.1.2$_b$
204 boundary nodes

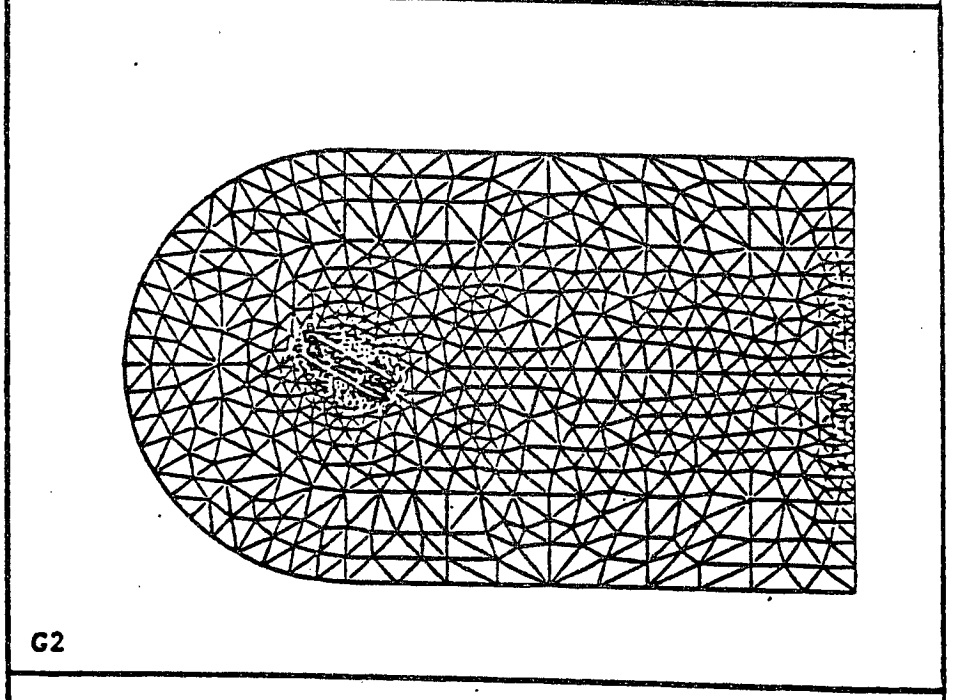


Fig. 5.1.2$_c$
276 boundary nodes

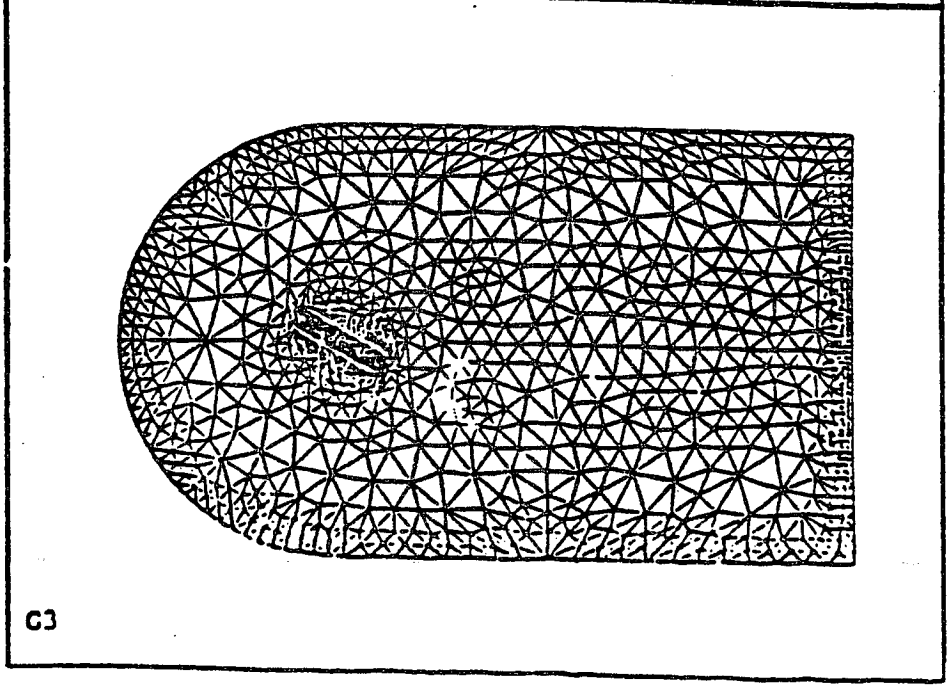


Fig. 5.1.3 - Influence of the grid
Stokes problem (α = 10., ν = 0.005)

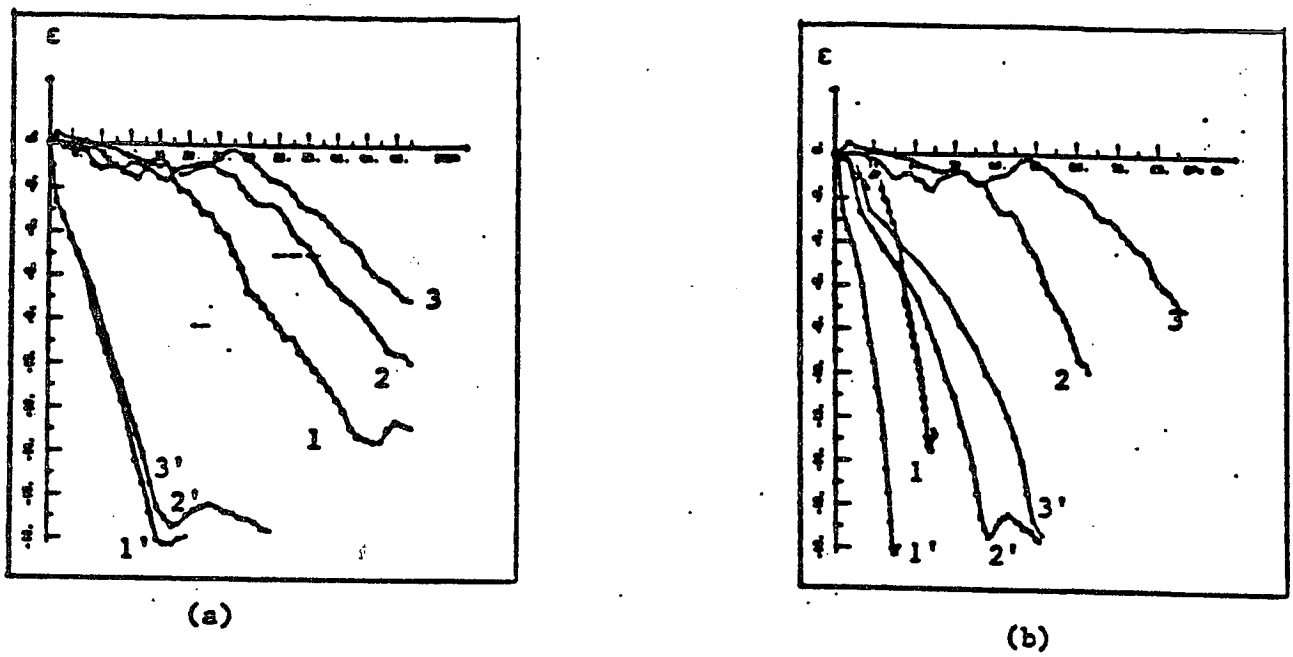


Fig. 5.1.4 - Influence of on the preconditioning
(Navier-Stokes problem)

1. α = 10. ν = 1. (1' with S)
2. α = 10. ν = 0.005 (2' with S)

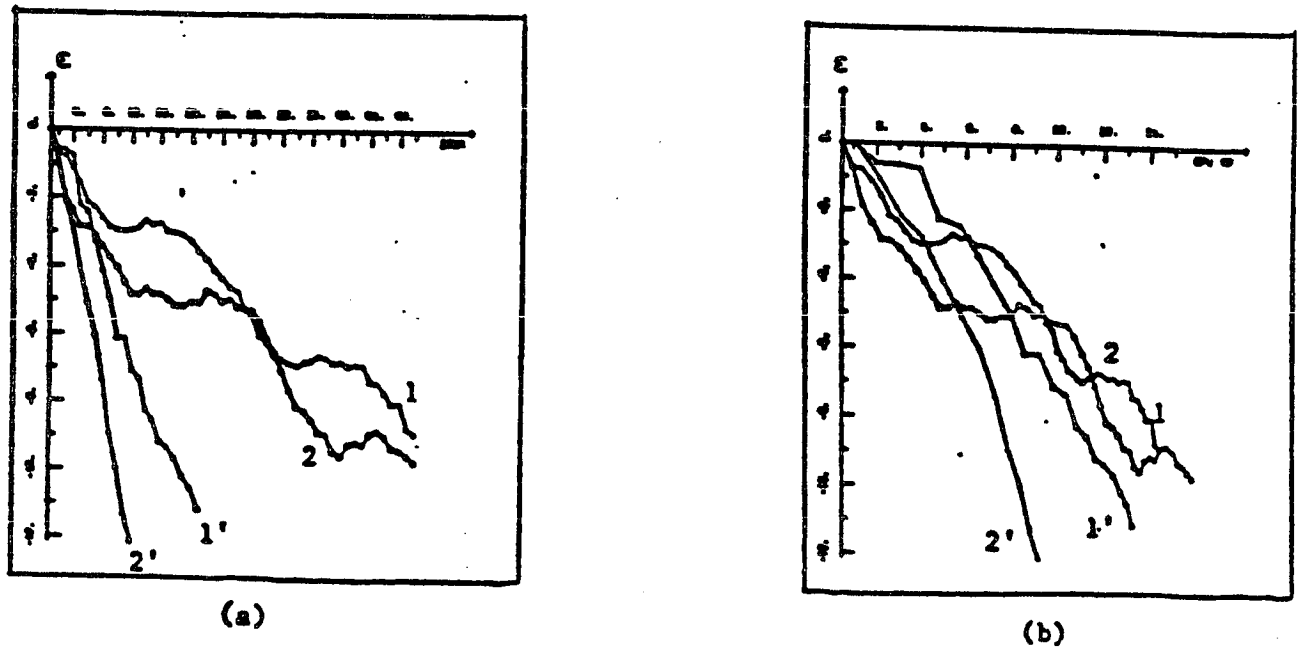


Fig. 5.2.2 - θ-schemes (P1/P1 - mini element)

3-D Navier-Stokes equations - Flow around a sphere

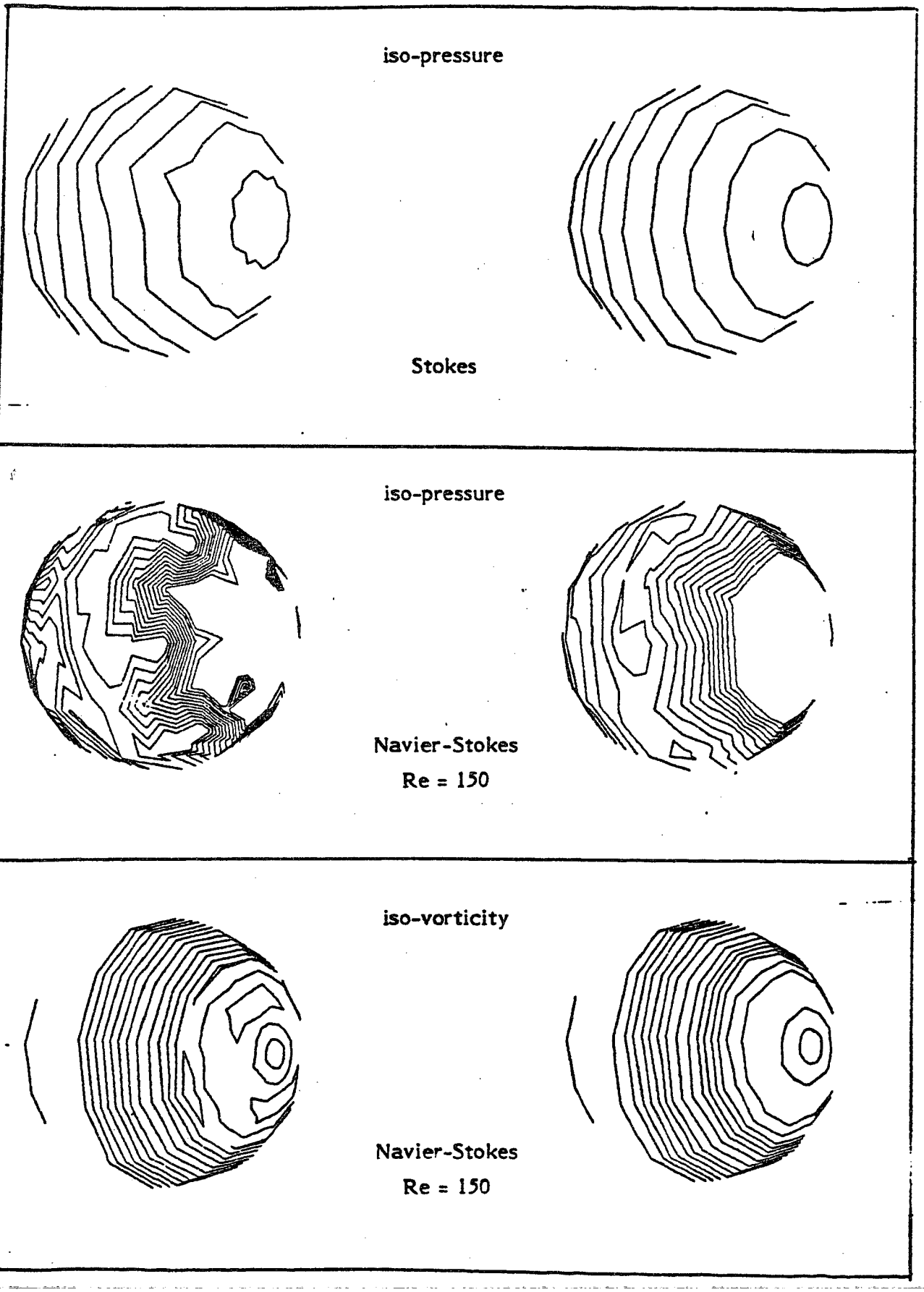
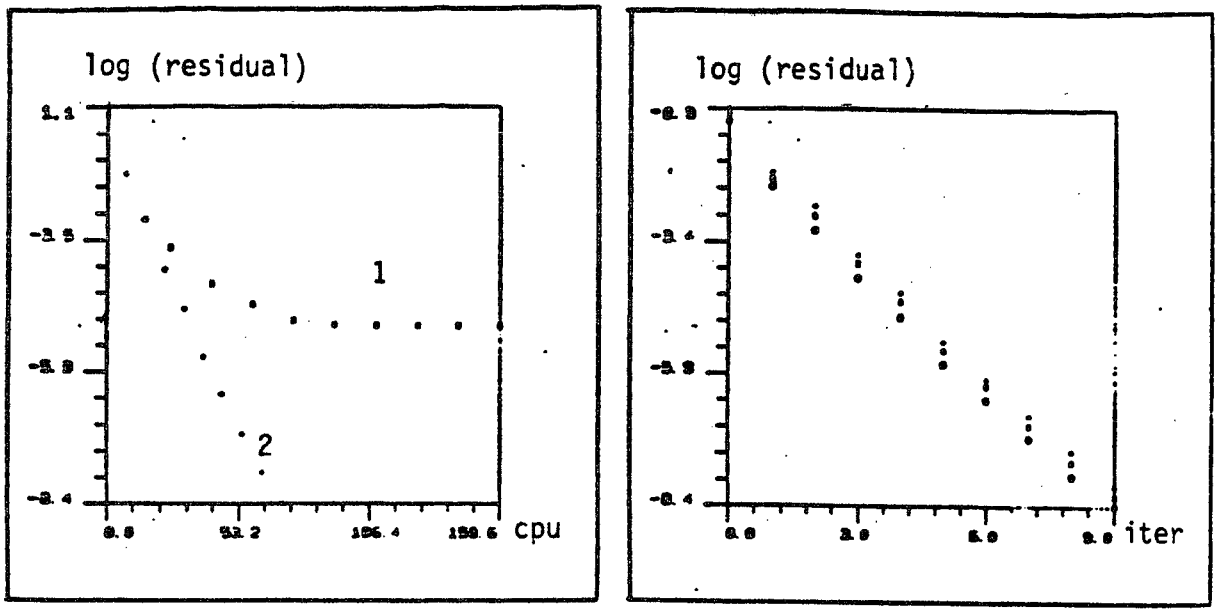


Fig. 5.3.1 - A multigrid method for solving
the Stokes problem (5.57)



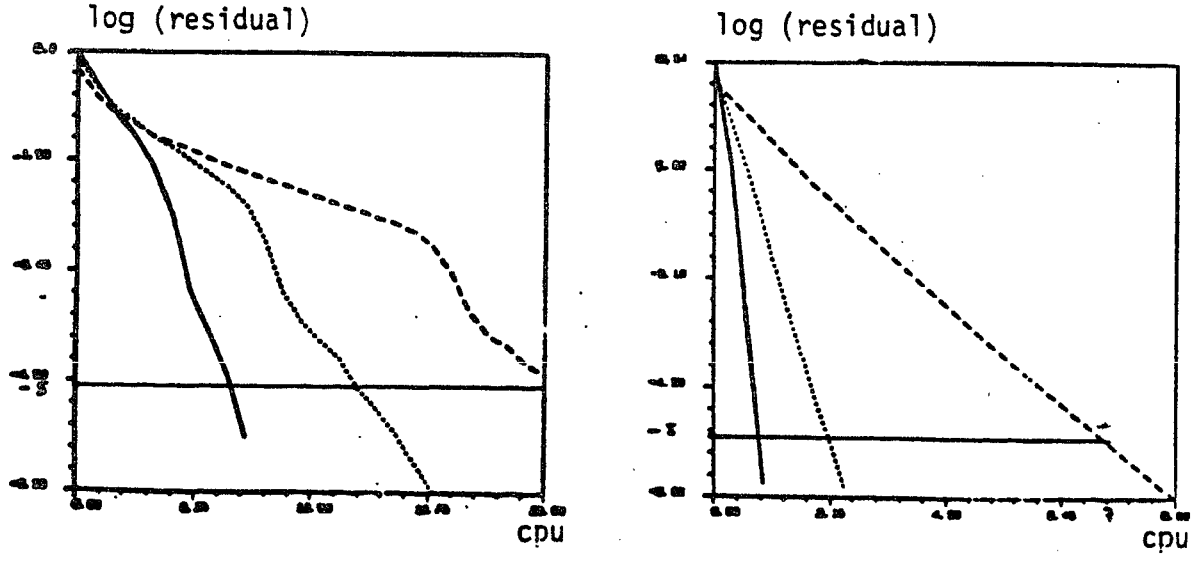
1. Uzawa
2. Multigrids

(a)

\+ 3-grids
* 4-grids
o 5-grids

(b)

Fig. 5.3.2 - A multigrid method for solving the Poisson problem
occuring in the nonlinear subproblem (5.58)



——— 81 nodes (G1)
..... 289 nodes (G2)
------1089 nodes (G3)

(a) Conjugate gradient

——— 3-grids
..... 4-grids
------- 5-grids

(b) Multigrids

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique